Lecture Notes in Artificial Intelligence 3029

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Bob Orchard   Chunsheng Yang
Moonis Ali (Eds.)

Innovations in Applied Artificial Intelligence

17th International Conference on
Industrial and Engineering Applications of
Artificial Intelligence and Expert Systems, IEA/AIE 2004
Ottawa, Canada, May 17-20, 2004
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Bob Orchard
Chunsheng Yang
National Research Council of Canada
Institute for Information Technology
1200 Montreal Road, M-50, Ottawa, ON, K1A 0R6, Canada
E-mail: {Bob.Orchard,Chunsheng.Yang}@nrc-cnrc.gc.ca

Moonis Ali
Texas State University-San Marcos
Department of Computer Science
Nueces 247, 601 University Drive, San Marcos, TX 78666-4616, USA
E-mail: ma04@txstate.edu

Library of Congress Control Number: 2004105117

CR Subject Classification (1998): I.2, F.1, F.2, I.5, F.4.1, D.2, H.4, H.2.8, H.5.2

ISSN 0302-9743
ISBN 3-540-22007-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by PTP-Berlin GmbH
Printed on acid-free paper SPIN: 11000969 06/3142 5 4 3 2 1 0

Preface

"Intelligent systems must perform in order to be in demand."

Intelligent systems technology is being applied steadily in solving many day-to-day problems. Each year the list of real-world deployed applications that inconspicuously host the results of research in the area grows considerably. These applications are having a significant impact in industrial operations, in financial circles, in transportation, in education, in medicine, in consumer products, in games and elsewhere. A set of selected papers presented at the seventeenth in the series of conferences on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004), sponsored by the International Society of Applied Intelligence, is offered in this manuscript. These papers highlight novel applications of the technology and show how new research could lead to new and innovative applications. We hope that you find these papers to be educational, useful in your own research, and stimulating.

In addition, we have introduced some *special sessions* to emphasize a few areas of artificial intelligence (AI) that are either relatively new, have received considerable attention recently or perhaps have not yet been represented well. To this end, we have included special sessions on e-learning, bioinformatics, and human-robot interaction (HRI) to complement the usual offerings in areas such as data mining, machine learning, intelligent systems, neural networks, genetic algorithms, autonomous agents, natural language processing, intelligent user interfaces, evolutionary computing, fuzzy logic, computer vision and image processing, reasoning, heuristic search, security, Internet applications, constraint satisfaction problems, design, and expert systems.

E-Learning

With its ability to reduce operating costs and train more people, e-learning is an attractive option for companies that are trying to balance business and educational goals. Information technology (IT) is rapidly changing the landscape of e-learning with the advent of new intelligent and interactive on-line learning technologies, multimedia electronic libraries, collaborative communities and workspaces, and improving knowledge sharing and education practices.

In particular, with the rapid development of the Internet and the World Wide Web, university and college programs offered in distributed e-learning environments are an alternative form of education for those students who are best served by flexible location and time schedules. The situation in which distance education is primarily used in selective situations to overcome problems of scale (not enough students in a single location) and rarity (a specialized subject not locally available) is being changed. The major trends of e-learning are *multi-mode integration*, *learner-centered environments*, and *service-oriented institutions.*

We selected for this special session a collection of outstanding papers highlighting the work of researchers and practitioners from academia and industry.

Human-Robot Interaction

Recently, humanoid robots such as Honda's ASIMO and Sony's QRIO or pet robots such as Sony's AIBO have become quite familiar and thus the symbiosis of robots and humans has become an exciting research area. Some of the many research topics being pursued include: expressive interaction with face/voice/gesture; spoken dialogue processing, dialogue modeling, user modeling, personality, and prosody; gesture recognition, face recognition, and facial expression; sound localization and visual localization; tactile and other sensory perception; and multi-modal integration of sensory information

At previous IEA/AIE conferences, low-level interactions were reported. However, this special session focuses on higher-level human-robot interactions. Through interactions with people, a humanoid robot recognizes the emotional states of a human by spoken dialogue or recognizes relationships between people and adapts its behaviors through a dynamic learning system. In addition, design methodology is discussed by observing human-robot interactions. We hope this special session will lead to more human-robot interaction research papers at IEA/AIE conferences.

Bioinformatics

Bioinformatics is an interdisciplinary research area, where computer scientists solve interesting and important problems in molecular biology by building models and manipulating huge amounts of data generated by biologists around the globe. The techniques being used by computer scientists include clever design of data structures and algorithms, machine learning, AI techniques and statistical methods. In the postgenome era, innovative applications of such techniques have been used to solve problems in molecular biology including protein-to-protein interaction, gene discovery and secondary structure prediction. We feel that it is time for the AI community as a whole to embrace bioinformatics with its challenging and interesting problems for the application of AI. Some general problem-solving methods, knowledge representation and constraint reasoning that were originally developed to solve industrial applications are being used to solve certain types of problems in bioinformatics and vice versa. Inclusion of bioinformatics as a special session enriched the conference and also provided an opportunity for other AI practitioners to learn about the ongoing research agenda of bioinformatics, which in turn may foster future collaboration among the participants of this conference.

Acknowledgements

A total of 208 papers from 28 countries were submitted for consideration this year. Of those, 129 (including 4 for the HRI special session, 6 for the bioinformatics special session and 9 for the e-learning special session) were accepted for publication. This required a large effort on the part of many people. We extend our sincerest thanks to all of the committee members, the reviewers and the NRC Conference Services staff for their contribution.

May 2004

Bob Orchard
Chunsheng Yang
Stan Matwin
Moonis Ali
Raja Loganantharaj
Hiroshi G. Okuno
Stephen Downs
Fuhua Lin

IEA/AIE 2004 Organization

General Chair: Ali, Moonis
Program Chair: Orchard, Bob
Program Co-chairs:
Yang, Chunsheng
Okuno, Hiroshi G.
Matwin, Stan

Special Session Chairs:
Lin, Fuhua Oscar
Downes, Stephen
Loganantharaj, Raja
Okuno, Hiroshi G.

Program Committee

Bai, Yun
Baumeister, Joachim
Buchanan, Bruce G.
Chi, Alvin Kwan
Chung, Paul
De Azevedo, Hilton J.S.
Del Pobil, Angel P.
Deugo, Dwight
Dini, Gion
Dobrowiecki, Tadeusz
Drummond, Christopher
Felferning, Alexander
Hendtlass, Tim
Ishizuka, Mitsuru
Ito, Takayuki
Kampmann, Markus
Kobayashi, Tetsunori
Kumara, Sounder
Lao, Xiao
Leakek, David B.
Letourneau, Sylvain
Lin, Hong
Lingras, Pawan
Liu, Sandy
Liu, Youhe
Martin, Joel D.
Matthews, Manton
Monostori, Laszlo
Murphey, Yi Lu
Nguyen, Ngoc Thanh
Prade, Henri
Shaheen, Samir I.
Sharma, Satish C.
Shih, Timothy K.
Stell, John
Tam, Vincent
Terano, Takao
Tounsi, Mohamed
Turney, Peter
Tzafestas, Spyros
Widmer, Gerhard
Wylie, Robert H.
Xu, Yuefei
Yamaguchi, Takahira
Yang, Shaohau H.
Yang, Lili
Zhang, Liang
Zhu, Xingquan
Zhue, Jane

External Reviewers

Ahriz, Hatem
Ally, Mohamoud
Andrey, Ptitsyn
Bahri, Parisa A.
Barrière, Caroline
Bouguila, Nizar
Bruhn, Russel
Cattral, Rob
Chen, Zaiping
Cheng, Gordon
Danilowicz, Czeslaw
De Bruijn, Berry
Debenham, John
Deogun, Jitender
Emond, Bruno
Esmahi, Larbi
Famili, Abolfazl
Farley, Benoit
Fournier, Hélène
Galitsky, Boris
Gorodnichy, Dmitry
Hamilton, Howard
Hashida, Koiti
Hennessy, Daniel N.
Hernandez-Orallo, Jose
Hinde, Chris J.
Holt, Peter
Hsuan, Ming
Japkowicz, Nathalie
Jennings, Steven F.
Jonathan, Wren
Kaikhah, Khosrow
Kark, Anatol
Katagiri, Yasuhiro
Kawamoto, Kazuhiko
Kim, Dohoon
Kiritchenko, Svetlana
Korba, Larry W.
Kunert, Klaus-Dieter
Kuniyoshi, Yasuo
Lin, Maria
Lourens, Tino
Marsh, Stephen P.
Milosavljevic, Aleksandar
Mouhoub, Malek
Nakadai, Kazuhiro
Ogata, Tetsuya
Ouazzane, Karim
Pan, Youlian
Petriu, Emil
Phan, Sieu
Picard, Rosalind W.
Ravindrakumar, Vinay
Regoui, Chaouki
Shen, Weiming
Shih, Timothy K.
Shu, Chang
Sobecki, Janusz
Song, Ronggong
Sugimoto, Akihiro
Trutschl, Marjian
Tsui, Kwoc Ching
Umeda, Kazunori
Valdes, Julio
Vigder, Mark
Wada, Toshikazu
White, Tony
Williams, Phil
Yee, George O.
Yim, Julian
Zhang, Haiyi
Zhang, Xia

Table of Contents

Session 1a: Neural Networks (1)

Session 1b: Bioinformatics (1)

Session 1c: Data Mining (1)

Session 2a: Neural Networks (2)

Session 2b: Bioinformatics (2)

Session 2c: General Applications

Session 3a: Autonomous Agents (1)

Session 3b: Intelligent Systems (1)

Session 3c: Knowledge Processing and Natural Language Processing

Session 4a: Intelligent User Interfaces

Session 4b: Evolutionary Computing (1)

Session 4c: Fuzzy Logic

Session 5a: Human Robot Interaction

Session 5b: Computer Vision and Image Processing

Session 5c: Machine Learning and Case Based Reasoning

Session 6a: Heuristic Search

Session 6b: Evolutionary Computing (2)

Session 6c: Security

Session 7a: Internet Applications

Session 7b: Planning and Scheduling

Session 7c: Constraint Satisfaction

Session 8a: E-learning (1)

Session 8b: Intelligent Systems (2)

Session 8c: Expert Systems

Session 9a: E-learning (2)

Session 9b: Applications to Design

Session 9c: Machine Learning

Session 10a: E-learning (3)

Session 10b: Autonomous Agents (2)

Session 10c: Neural Networks (3)

Session 11a: Data Mining (2)

Session 11b: Intelligent Systems (3)

Session 11c: Neural Networks (4)

Session 12a: Image Processing

Session 12b: Evolutionary Computing (3)

Session 12c: Data Mining (3)

A Comparison of Neural Network Input Vector Selection Techniques

Belinda Choi[1], Tim Hendtlass[1], and Kevin Bluff[2]

[1] Department of Information Technology, La Trobe University, Bendigo
b.choi@latrobe.edu.au
[2] School of Information Technology, Swinburne University of Technology, Melbourne
thendtlas@swin.edu.au, kbluff@swin.edu.au

Abstract. One of the difficulties of using Artificial Neural Networks (ANNs) to estimate atmospheric temperature is the large number of potential input variables available. In this study, four different feature extraction methods were used to reduce the input vector to train four networks to estimate temperature at different atmospheric levels. The four techniques used were: genetic algorithms (GA), coefficient of determination (CoD), mutual information (MI) and simple neural analysis (SNA). The results demonstrate that of the four methods used for this data set, mutual information and simple neural analysis can generate networks that have a smaller input parameter set, while still maintaining a high degree of accuracy.

Keywords: Artificial neural network, genetic algorithm, mutual information, coefficient of determination.

1 Introduction

Temperature estimation is an important component of weather forecasting. Forecasters at the Australian Bureau of Meteorology currently use a number of methods to assist them in making such estimations. The TIROS Operational Vertical Sounder (TOVS) instrument is used to observe radiation from the earth and atmosphere. Conventional methods used to retrieve temperature and moisture profiles from the observed TOVS radiances are iterative and involve matrix algebra, making them very computationally intensive. In general, both physical and statistical retrieval techniques show differences of about 2 K when compared to analysis or collocated radiosonde data in the mid-troposphere (600 – 400 hPa). These differences often increase near the tropopause and surface of the atmosphere [1].

Artificial Neural Networks (ANNs) are good functional approximators and are potentially well suited to the task of temperature estimation. One of the main features of ANNs is their ability to generalise; that is, to successfully interpolate between examples previously seen. Moreover, once trained, ANNs can run at a much higher speed than iterative matrix methods. A task that is critical to the success of a neural network in weather forecasting problems is the selection of inputs. At each particular time instant, values exist for a vast number of meteorological variables available including infrared and visible satellite readings, ground measurements, humidity, wind speed and

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1-10, 2004.

direction at many different pressure levels in the atmosphere. Use of an extremely large number of inputs to an ANN makes training the network considerably more difficult and sharply increases the amount of training data required.

A number of different methods have been used to reduce the size of the input vector of an ANN. These include correlations between candidate input parameters and the modelled parameter, schematic analyses and the use of expert advice. Some of these methods are subjective and do not measure the multivariate dependencies present in the system [2][3]. In this study, genetic algorithms (GA), coefficient of determination (CoD), mutual information (MI), simple neural analysis (SNA) and expert knowledge (EK) are used for input vector reduction and compared.

2 Genetic Algorithms

Genetic Algorithms (GAs) [4] are search algorithms based on the theory of natural selection. In this application, each individual (chromosome) in a population is a string containing 0's and 1's (genes) and describes one possible selection of inputs. Successive populations are generated using a breeding process that favours fitter individuals. The fitness of an individual is considered a measure of the success of the input vector. Individuals with higher fitness will have a higher probability of contributing to the offspring in the next generation ('Survival of the Fittest').

There are three main operators that can interact to produce the next generation. In replication individual strings are copied directly into the next generation. The higher the fitness value of an individual, the higher the probability that that individual will be copied. New individuals are produced by mating existing individuals. The probability that a string will be chosen as a parent is fitness dependent. A number of crossover points are randomly chosen along the string. A child is produced by copying from one parent until a crossover point is reached, copying then switching to the other parent and repeating this process as often as required. An N bit string can have anything from 1 to N-1 crossover points.

Strings produced by either reproduction or crossover may then be mutated. This involves randomly flipping the state of one or more bits. Mutation is needed so new generations are more than just a reorganisation of existing genetic material. After a new generation is produced, each individual is evaluated and the process repeated until a satisfactory solution is reached. Because of the need for large populations and a substantial number of generations, the number of evaluations may be high.

3 Correlation Coefficient and Coefficient of Determination

Pearson's correlation is a measure of the degree of linear relationship between these two variables [5].

The coefficient of determination (CoD) is the square of the correlation coefficient and can only have positive values ranging between 1 for a perfect correlation (whether positive or negative) to 0 for a complete absence of correlation. The CoD gives the percentage of the explained variation compared to the total variation of the model. For example, if the CoD between X & Y is 0.55 (55%), it can be said that 55% of the variability of X is explained by the variability in Y [5].

By computing the CoD between each input / output pair and selecting only highly correlated inputs for each output, one may hope to select a good, if not optimal, set of inputs. However, this will only select an effective reduced input set if all the inputs are independent and no output is in fact a complex function of two or more of the input variables – an assumption that cannot always be relied upon.

4 Mutual Information between Inputs and Outputs

Mutual information (MI) [6] is another measure of the amount of information that one variable contains about another variable. While the coefficient of determination measures the linear dependence between two variables, mutual information measures the general dependence between two variables and is therefore potentially a better predictor of dependence than the correlation function [7].

The mutual information of independent variables is zero, but is large between two strongly dependent variables with the maximum possible value depending on the size of the data set.

Mutual information can, in principle, be calculated exactly if the probability density function of the data is known. Exact calculations have been made for the Gaussian probability density function [7]. However, in most cases the data is not distributed in a fixed pattern and the mutual information has to be estimated. In this study the mutual information between each input and each output of the data set is estimated using Fraser & Swinney's method [8].

A subset of inputs with high mutual information to the relevant output is selected to train the neural network. Again this assumes that all the inputs are independent and that no output is in fact a complex function of two or more of the input variables.

5 Simple Neural Analysis

Simple neural analysis (SNA) attempts to select the most appropriate system inputs to the final network using either single neurons or networks of neurons. In this paper only N input, 1-output neurons were used (where N was either 1 or 2). Either 1 or 2 of the system inputs under consideration were used to train the neuron to predict a particular system output. After training, the CoD between the actual neuron output and the ideal system output provides a measure of the importance of these system input(s) to this output. SNA is the only technique of those described in this paper that can estimate the importance of multiple inputs in a single step. Obviously N cannot be allowed to become large or combinational explosion occurs. A series of neurons were also trained using one system input as input and a second system input as the output so as to identify high related system inputs. System inputs that had a high CoD to one or more system outputs and which were unrelated to any other system input were considered as good candidates for use as inputs to the final network.

6 Expert Knowledge

An expert in the field (EK) selected N inputs that were believed to make a significant contribution. It was considered difficult to rank the inputs accurately.

7 Methodology

The temperature data for Australia was taken from the TOVS instrument equipped NOAA12 satellite in 1995. Infrared sounding of 30km horizontal resolution was supplemented with microwave soundings of 150 km horizontal resolution.

This data set was used to evaluate the techniques for selecting the input vector. A number of single output networks were developed, each estimating the actual temperature at one of 4 pressure levels (1000, 700, 300 & 150 hPa) given the radiances measured by satellite. These are four of the standard pressure levels (levels 1, 3, 6 and 9) measured by satellite and radiosonde sounders. The input vector of TOVS readings to be used by these networks was extracted using each of the five techniques described above. A common input vector length of 8 was used as initial experimentation had proved this to be a suitable value. The appropriate target output temperature was provided by collocated radiosonde measurement.

In general each input was tested against the output for importance and a ranked list was produced then the top 8 inputs were selected for the input vector. Experimental details for the four methods used to rank these inputs above are:

- Coefficient of Determination and Fraser and Swinney's mutual information estimation method involve no user selectable parameters.
- In Simple Neural Analysis, a 1-input-1-output architecture was used and training continued for 100 passes through the data set with a learning rate of 0.1 and a momentum of 0.8. These values were chosen based on prior experimentation.
- This general process was not applicable to the GA technique where, as indicated in Section 2, the input vector was directly represented by a chromosome encoding which was then subjected to evolutionary pressure. The fitness function used was simply E, the percent correct ($|\text{target - actual}| < 0.4$). To measure the number correct for a given chromosome 5 networks were trained using these inputs and the best result used. The 8-12-1 network uses a learning rate of 0.1 and momentum of 0.8 for 10,000 iterations.
- Parents were selected using a ranked performance order and an elitist strategy. The GA bred 20 generations with a population size of 50 using a crossover rate of 0.6 and a mutation rate of 0.05. It was found that there was very little increase in performance after 20 generations. Although the inputs selected by 5 GA instances (with different random seeds) were not identical, there was enough consistency to confidently select the set of results that gave the lowest RMS error. The input vector selected by the GA was evaluated in the same general manner described above.
- After selecting 8 best inputs using one of the above techniques, these inputs were assessed by means of an evaluation neural network whose architecture was chosen based on initial experiments. The network used 12 hidden neurons and was trained using fixed parameters to facilitate comparison between the various techniques. It was trained for 2000 passes through the data set using a learning rate of

0.1 and a momentum of 0.8. The network was tested after each pass though the training data with the best result being recorded. The overall performance of this testing network was assumed to reflect the appropriateness of this particular selection of inputs.

8 Results and Discussion

The results reported are the median RMS error values (degrees Kelvin) obtained from training the ten evaluation networks at each level and should be a reasonable reflection of the inherent worth of the input selection. The results using the eight inputs predicted by the expert (EK) and using the full input vector (all available inputs) are included in the table for comparison. The spread of RMS errors produced by the ten evaluation networks is also a performance indicator and is indicated in Table 1 by including the highest and lowest errors as + and – values relative to the median.

Table 1. Median and range of RMS Error (K) using the 8 inputs derived from all 5 techniques and using all inputs for Levels 1,3 6 & 9.

	CoD	EK	GA	MI	SNA	ALL
Level 1	$3.1^{+0.2}_{-0.4}$	$3.8^{+0.0}_{-0.2}$	$3.3^{+0.2}_{-0.1}$	$2.7^{+0.1}_{-0.1}$	$2.5^{+0.1}_{-0.0}$	$2.9^{+0.1}_{-0.1}$
Level 3	$3.5^{+0.2}_{-0.1}$	$3.6^{+0.3}_{-0.0}$	$3.4^{+0.1}_{-0.1}$	$3.6^{+0.0}_{-0.1}$	$3.1^{+0.2}_{-0.2}$	$2.7^{+0.0}_{-0.1}$
Level 6	$2.5^{+0.2}_{-0.0}$	$2.6^{+0.3}_{-0.1}$	$2.8^{+0.3}_{-0.2}$	$2.4^{+0.0}_{-0.0}$	$2.4^{+0.8}_{-0.0}$	$2.6^{+0.3}_{-0.5}$
Level 9	$4.0^{+0.0}_{-0.1}$	$3.5^{+0.1}_{-0.1}$	$3.8^{+0.0}_{-0.2}$	$3.4^{+0.1}_{-0.0}$	$3.4^{+0.1}_{-0.1}$	$3.9^{+0.1}_{-0.2}$

Table 1 indicates that MI and SNA exhibited fairly similar performance and were generally better than the other three-vector reduction techniques at most levels. SNA was marginally better than MI, outperforming it in levels 1 and 3. In three of the four levels both MI and SNA managed to improve network performance, compared with using all inputs, as well as reduce the input vector.

Level 3 is interesting in that all five reduction techniques produced networks with worse performance. An indication of the underlying cause of this phenomenon is that for this level the individual importance measures of the inputs for all three quantitative techniques (EK and GA provide no such individual measure) were at best 50% lower and at worst two orders of magnitude lower than for the other levels. This seems to indicate that the predictive capability at this level is spread more across the inputs – there is less redundancy of information. It should be noted that at this difficult level SNA outperformed the other four techniques, doing less “harm”.

A comparison of the individual inputs selected for inclusion by the five techniques can provide some insight into their manner of operation and the broad relative importance of the inputs. Table 2 and Figure 1 provide this comparison. As described previously only CoD, MI and SNA actually rank the inputs and thus select them in an ordered manner with EK and GA simply specifying the complete set. In Table 2 the latter two techniques have their inputs presented in numeric order and in Figure 1 they have all inputs indicated as of ‘equal importance’.

One feature of Table 2 which is of immediate interest is that although there is considerable similarity between the inputs selected by MI and SNA there are substantial differences across the techniques. Curiously, the GA selected the same input set for all but level 1 which was considerably different, and still attained comparable performance. Perhaps these differences indicate considerable underlying information redundancy within the data set as various combinations of inputs can be used with some success.

Table 2. The 8 inputs derived from the 5 vector reduction techniques for Levels 1,3 6 & 9 in order of inclusion. (No order exists for EK and GA)

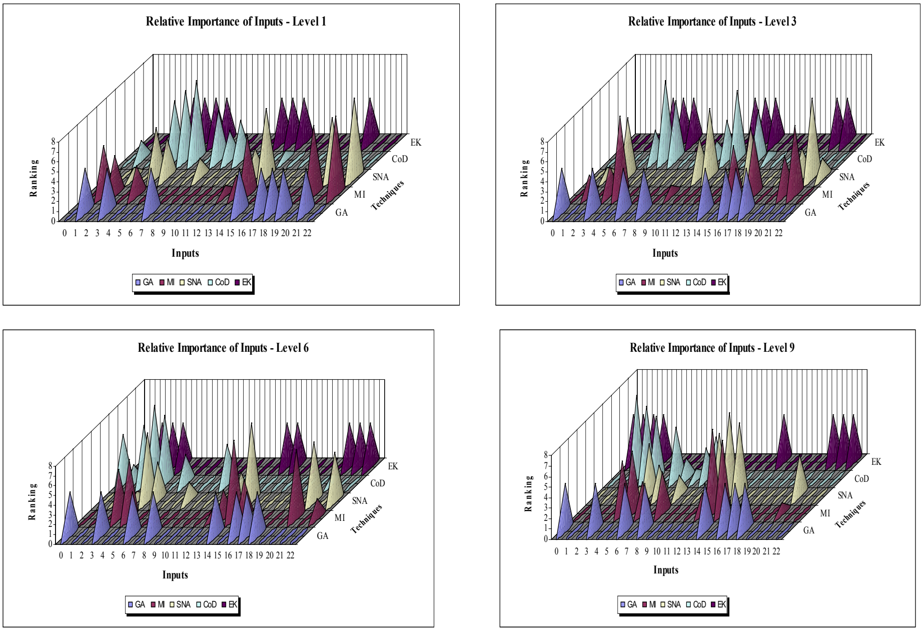
	CoD	**EK**	**GA**	**MI**	**SNA**
Level 1	6, 5, 4, 8, 10, 9, 1, 14	4, 5, 6, 7, 12, 13, 14, 20	1, 3, 7, 15, 17, 18, 19, 21	22, 20, 14, 1, 2, 4, 13, 12	22, 14, 20, 4, 5, 13, 8, 2
Level 3	5, 12, 6, 14, 11, 4, 10, 15	4, 5, 6, 12, 13, 14, 20, 21	0, 3, 6, 8, 14, 16,17, 18	4, 21, 17, 15, 20, 3, 1, 9	21, 11, 3, 15, 10, 20, 22, 1
Level 6	3, 4, 2, 0, 10, 6, 1, 20	2, 3, 4, 14, 15, 20, 21, 22	0, 3, 6, 8, 14, 16,17, 18	14, 20, 4, 3, 15, 13, 22, 12	14, 4, 20, 22, 5, 13, 8, 3
Level 9	2, 3, 4, 6, 10, 9, 7, 8	0, 1, 2, 3, 15, 20, 21, 22	0, 3, 6,8, 14, 16,17, 18	13, 14,5, 4, 8, 6, 12, 20	13, 14, 12, 5, 20, 6, 8, 4

The similarity in input sets selected by the MI and SNA techniques is highlighted in Table 3. Given the similar performance of these techniques it is not surprising that there is considerable overlap.

Table 3. Inputs selected by MI & SNA for various levels.

Level	MI only	Common	SNA only
1	1, 12	2, 4, 13, 14, 20, 22	5,8
3	4, 9, 17	1, 3, 15, 20, 21	10, 11, 22
6	12, 15	3, 4, 13, 14, 20, 22	5, 8
9		4, 5, 6, 8, 12, 13, 14, 20	

Only input 20 is common at all levels in Table 3 and this input is also the only one common to all levels in the selections made by the expert (EK, Table 1) thus there seems to be broad agreement on its importance. This also seems reinforced by its absence from the poorly performing selections made by CoD at levels 1, 3 and 9. Indeed, the only level (6) at which CoD's selection performs well at all has this input included. It would seem that this highlights the inappropriateness of a model with an underlying linear bias for this task.


Relative Importance of Inputs - Level 1

Relative Importance of Inputs - Level 3

Relative Importance of Inputs - Level 6

Relative Importance of Inputs - Level 9

Fig. 1. Relative Importance of inputs using the 5 different techniques (the higher the peaks, the more important the input). Level 1 top left, level 3 top right, level 6 bottom left, level 9 bottom right.

The results obtained above imply that both MI and SNA have some considerable potential for input vector reduction. Further experiments were conducted for these techniques to investigate the possibility of further pruning the number of inputs. The best 4 inputs were used instead of 8 and the process repeated using 10 evaluative networks at different starting positions. The results were surprisingly good, with a small general degradation in performance as measured by median RMS error.

Multiple Input SNA. As was noted earlier, an assumption underlying the CoD, MI and SNA techniques as they have been used so far is that the inputs are essentially independent and that no output is in fact a complex function of two or more of the input variables. Of the three techniques SNA is the only one that has the capacity to directly take into account more than one input. By increasing the number of inputs to the simple neural processing unit, simple combinations of two or more inputs can be related to the output. The investigation here was restricted to two-input combinations because combinatorial explosion soon dominates. All 253 two-input combinations were trained to predict each output and ranked along with the single input results.

Selecting a fixed number of inputs from a ranked list consisting of combinations along with single entries is somewhat problematical. When considering a ranked entry which is a combination (in this investigation a pair) it is necessary to ascertain as far as possible whether or not one of its components is the dominant "information carrier", with the other component(s) contributing minimally, or whether it is truly the combination which is important. If one component is truly dominant and both components are added, the resultant vector could be reduced in effectiveness because part of its fixed capacity has been used on an input of only peripheral value, potentially excluding more useful inclusions. The approach adopted in this investigation was very simplistic: the ranked list is processed from the highest scoring entry downwards and a decision is made with respect to each entry as follows as long as the fixed-length input vector is not full;

1. if the entry is a single input and is not present in the vector add it,
2. if the entry is two inputs not present in the vector add both if the combination significantly outperforms the single input entries for both components, otherwise add the dominant (higher performing) one,
3. if the entry is two inputs with one already present in the vector add the other only if the combination significantly outperforms that one's single input entry.

For this study, one entry is considered to 'significantly outperform' another if it has a CoD value greater by 0.1 or more.

Although the application of this simplistic attempt to explicitly take account of the potentially important interactions between inputs did alter the input vector components, the only significant effect on performance was to reduce the performance at levels 1 and 3 to that attained by the initial MI technique. More investigation is required to develop an effective way to use the extra information available from input combinations.

Redundancy in the Input Set. The previous section discussed the potential for relationships between inputs making combinations that were more useful members of the reduced input vector than sum of their individual components. Another relationship which can exist is that two or more inputs can be carrying very similar information (essentially measuring the same thing in a different way). Whilst each input may individually have a very high ranking it can be counter-productive to add both. Two versions of the same information would be occupying space in the fixed length input vector with little increase in performance but potentially excluding a lower ranked input which adds extra information and does increase performance. A simple investigation attempted to discover if the selection process could be improved by taking this into account.

Relationships between all single-input-single-input combinations were tested with three techniques, SNA, CoD & MI (of course this analysis was not possible with EK and GA) and selection of the 8 optimal inputs was adjusted to eliminate inputs which are highly interrelated. For SNA two-input-single-input relationships were also evaluated. The selection process simply added another test for inclusion in the reduced input vector. If the potential addition was deemed 'highly related' to something already included then it was not added. The definition of 'highly related' for this purpose is somewhat problematical and values for each of the relationship measures were established after some initial investigation.

Unfortunately this approach was probably too simplistic (or the definition values used too conservative) as the performance exhibited by the newly chosen reduced vectors was generally a little worse for SNA and MI. The CoD performance was marginally improved, but was starting from a worse point. Further investigation of how to define 'highly related' and how to incorporate this information into the selection process is indicated.

9 Conclusion

In this paper, GA. MI, CoD and SNA are used to select the best inputs to neural networks. In order to be able to compare the 4 methods, 8 best inputs were selected for training of the evaluation neural network. The MI and SNA techniques gave the best results overall. The results indicate that reducing the input vector using these techniques not only makes training the neural networks more efficient but also that this may be achieved without significant loss of performance. The exception is Level 3 (700 hPa) where using all inputs provided the best performance. A possible general indicator of when this might occur lies with the observation that all the individual input-output relationship measures for this level were considerably lower.

The facts that the performance did not dramatically deteriorate when the input vector was further reduced to contain only four inputs, and that the ‘strange’ selections of the GA still attained some success, both seem to indicate that there is some redundancy in the information carried by the full input set. The lack of success of the attempts to incorporate information on multiple-input combinations and input redundancy is intriguing and bears further investigation and refinement of the simple techniques employed.

Details of the TOVS instrument system and the purpose of the radiance observation as documented in [9] shows that HIRS channels 0-6, 12 – 16 and MSU channel 2 -4 are channels that provide better sensitivity to the temperature sounding. HIRS channels 0-6 have better sensitivity to the temperature of relatively cold regions of the atmosphere, HIRS 12 – 16, a better sensitivity to the temperature of relatively warm regions of the atmosphere and also less sensitive to clouds. MSU channels 2 – 4 have the microwave channels which probe through clouds and can be used to alleviate the influence of clouds on the other channels. This information indicates that, for the levels considered in this paper, channels 0, 1, 2, 3, 4, 5, 6, 12, 13, 14, 15, 16, 20, 21 & 22 will be expected to contribute more towards temperature estimation than the other channels. Comparing this with the 8 best inputs selected by each technique, it is interesting to note the EK, SNA & MI techniques comply very closely with this observation. The GA and CoD techniques are the least compliant with a total of at least 10 inputs selected for the 4 levels that do not fall in this range. It is also interesting to note for further investigation that only the GA selected input 0 with both the SNA and MI techniques giving it a very low ranking.

Although this investigation has been conducted only on one set of data, these techniques show very promising results and may well be successfully applicable to other data sets. Obviously future work in this area would include further investigations as outlined above and extending the application of these techniques to different data sets in order to gauge their usefulness in a more general context. The ultimate goal is a range of techniques that can be applied in a disciplined and fully automated fashion to any data set to reduce the size of the input vector a neural network will use to estimate or predict.

Acknowledgements. Thanks to the CISCP, Swinburne University of Technology for providing the simple neural analysis (SNA) software, John LeMarshall from the Bureau of Meteorology for the use of the real-time temperature data and expert knowledge information; and Fraser & Swinney for the use of their code for the estimation of mutual information.

References

[1] J. LeMarshall. An Intercomparison of Temperature and Moisture Fields Derived from TIROS Operational Vertical Sounder Data by Different Retrieval Techniques. Part I: Basic Statistics. *Journal of Applied Meteorology*, Vol 27, pp1282 – 1293, 1988.

[2] A. Bowles. Machine Learns Which Features to Select. *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pp 127-132, 1992.

[3] J.Nadal, N.Brune, & N.Parga. Nonlinear feedforward networks with stochastic outputs: infomax implies redundancy reduction. *Network: Computation in Neural Systems,* 9(2), pp207-217. 1998.

[4] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley. 1989.

[5] N. Weiss & M. Hassett,. *Introductory Statistics*, Addison-Wesley, USA. 1987

[6] W. Li,. Mutual Information Functions versus Correlation Functions. *Journal of Statistical Physics*, Vol 60, Nos.5/6, pp 823 - 836. 1990.

[7] F. Reza. *An Introduction to Information Theory*. McGraw-Hill Book Co., New York. 1961.

[8] A.M. Fraser & H.L. Swinney. Independent Coordinates for Strange Attractors from Mutual Information. *Physical Review A*, Vol. 33/2, pp 1134 – 1140. 1986.

[9] Bureau of Meteorology (1993). *Manual of Meteorology – Part 1. General Meteorology.* Australian Government Publishing Service, Canberra.

Application of Direction Basis Function Neural Network to Adaptive Identification and Control

Mahdi Jalili-Kharaajoo

Young Researchers Club
Azad University, Tehran
P.O. Box: 14395/1355, Tehran, Iran
mahdijalili@ece.ut.ac.ir

Abstract. In this paper, adaptive identification and control of nonlinear dynamical systems are investigated using Two Synaptic Weight Neural Networks (TSWNN). Firstly, a novel approach to train the TWSWNN is introduced, which employs an Adaptive Fuzzy Generalized Learning Vector Quantization (AFGLVQ) technique and recursive least squares algorithm with variable forgetting factor (VRLS). The AFGLVQ adjusts the kernels of the TSWNN while the VRLS updates the connection weights of the network. The identification algorithm has the properties of rapid convergence and persistent adaptability that make it suitable for real-time control. Secondly, on the basis of the one-step ahead TSWNN predictor, the control law is optimized iteratively through a numerical Stable Davidon's Least Squares-based (SDLS) minimization approach. A nonlinear example is simulated to demonstrate the effectiveness of the identification and control algorithms.

1 Introduction

Similar to the Multilayer Feedforward Neural Networks (MFNN), Two Synaptic Weight Neural Networks (TSWNN) [1] possesses the capacity of universally approximating nonlinear multi-variable functions [2]. Unlike the former, the output of the TSWNN is linear with respect to the connection two weights of the network. If the other parameters, the TSWNN kernels, can be chosen appropriately, the linear least squares method can therefore be employed to estimate these weights, so the rapid convergence of the algorithm will be guaranteed. Therefore, the performance of a TSWNN critically depends upon the chosen kernels.

Wang Shoujue [3] suggested that the kernels were randomly chosen from data points. This method is clearly simple but raw. An n-means clustering technique was regarded as a better method for updating the kernels by Feng Cao [4]. Wang [3] presented Slam algorithm for pattern matching, but it is too complex to match on-line application. For on-line and adaptive applications of neural network model, some kinds of recursive identification algorithms are naturally required [5,6,7,8]. A simple solution is to fix the kernels first using n-means clustering algorithm and to update only the two sorts of weights in real-time using recursive least squares or least mean squares algorithm. This can only work well if the variations in the underlying system are small. It is advantageous to update TSWNN kernels and weights simultaneously.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 11-19, 2004.

Therefore, Wang proposed a hybrid n-means clustering and Givens least squares algorithm. Given least squares have superior numerical properties.

In this paper, the kernels and weights of the TSWNN are also updated simultaneously. A novel approach, however, is presented to train the TSWNN. Specifically, an Adaptive Fuzzy Generalized Learning Vector Quantization (AFGLVQ) technique is adopted to update the kernels, its weights and recursive least squares algorithm with variable forgetting factor (VRLS) is used to estimate the weights. The FGLVQ is a fuzzy modification of the GLVQ clustering algorithm. Compared with n-means clustering and the GLVQ, this algorithm seems more insensitive to the disturbance of random initialization. Moreover, it more likely gives a desirable solution than the other algorithms, and it still keeps the computational simplicity. We enhance the adaptability of the FGLVQ by giving a new design for the learning rate. Because the clustering algorithms are originally designed for pattern recognition and a generic assumption is that a fixed cluster exists for finite data points, the learning rates are designed to slowly decrease to zero such that these algorithms can converge to a locally optimal solution. In identification and control of dynamical systems, the kernels should persistently reflect the variations of the input sequence. Therefore, the learning rate will be designed to vary proportionally to the identification error.

VRLS perfectly updates the parameters to follow both slow and sudden changes in the plant dynamics. Furthermore, the use of a variable forgetting factor with correct choice of information bound can avoid blowing-up of the covariance matrix of the estimates and subsequent unstable control. The computational load is relatively small. Of course, some recursive least squares algorithm with superior numerical properties can also be used under the consideration of moderate computational load.

Up to the present, the control schemes based on the TSWNN have not been as many as those based on the other networks such as MFNN, CMAC and recurrent networks. In this paper, another scheme of adaptive control law will be given. On the basis of the one-step ahead TSWNN predictor, a numerically stable Davidon's least squares-based (SDLS) minimization method is proposed to optimize the control law iteratively in each sample period. The DLS algorithm possesses relative fast learning property but suffers numerically ill-conditioned phenomenon. The present SDLS has overcome the drawback.

2 Two Synaptic Weight Neural Networks for Modeling Nonlinear System

Many single-input single-output non-linear systems can be described as the following model

$$y(t) = f_s(y(t-1),...,y(t-n_y),u(t-1),...,u(t-n_u)) \qquad (1)$$

where *y(t)* and *u(t)* are the system output and input respectively; n_y and n_u the lags of the output and input respectively; and $f_s(.)$ some non-linear function.

The TSWNN is a two-layer processing structure. In the first layer, the neurons are represented by TSWNN with the kernels c_i, which are interconnected by weights w_{ij}, w'_{ji}. Factor *s, p* is chosen. The second layer is essentially a linear combiner. The overall response of such a network is a mapping $f_r : R^m \rightarrow R$, that is

$$f_r(x) = \sum_{i=1}^{n} \theta_i \phi(x, c_i, w_i) \tag{2}$$

where $x \in R^m$, m is network input vector, $c_i \in R^M, 1 \le i \le n$ are the kernels and $\phi(.) : R \rightarrow R$. θ_i is the connection weights; and n is the number of neurons.
When the TSWNN is used to approximate the dynamical system (1), define *m= ny + nu* and let

$$x(t) = [y(t-1), ..., y(t-n_y); u(t-1), ..., u(t-n_u)] \tag{3}$$

Then, the TSWNN output

$$\hat{y}(t) = f_r(x(t)) \tag{4}$$

acts as the estimator of the *y(t).* Let

$$e(t) = \hat{y}(t) - y(t) \tag{5}$$

Hence, the goal of training the TSWNN is to make *e(t)* as small as possible. In this paper, the function is chosen as function

$$\phi(x, c_i, w_i) = \cos\left[\sum_{j=1}^{m}\left(\frac{w_j}{|w_j|}\right)^s |w_j(X_j - c_j)|^p - \theta\right] \tag{6}$$

X_j is neuron j^{th} input; w_j, c_j are direct weight and kernel weigh; θ is the threshold; m is the dimension of input.

3 Hybrid AFGLVQ and VRLS Algorithm

The adaptive identification algorithm for TSWNN model has a hybrid structure consisting of

1. recursive AFGLVQ sub-algorithm for adjusting the TSWNN kernels;
2. recursive VRLS sub-algorithm for updating the TSWNN weights;

Now, we begin to give the details of the two sub-algorithms.

3.1 Adaptive FGLVQ

3.1.1 FGLVQ

In the GLVQ algorithm, for the winner node *i(i= arg min*| *x(t)-c*$_j$*(t)*|*, 1< i< n)*, the updating rules of the kernels and its weights are

$$c_i(t+1) = c_i(t) + b(t)\left[\frac{D^2 - D + \|x(t) - c_i(t)\|}{D^2}\right](x(t) - c_i(t))$$
$$w_i(t+1) = w_i(t) + \eta_o \|x(t) - c_i(t)\|(x(t) - c_i(t)) \tag{7}$$

For the rest nodes *j (j=1,..., n,* $j \neq i$ *),* the rules are

$$c_j(t+1) = c_j(t) + b(t)\left[\frac{D^2 - D + \|x(t) - c_j(t)\|}{D^2}\right](x(t) - c_j(t))$$
$$w_j(t+1) = w_j(t) + \eta_o \|x(t) - c_j(t)\|(x(t) - c_j(t)) \tag{8}$$

where $D = \sum_{i=1}^{n} \|x(t) - c_i(t)\|$, η_o, η_1 1 is proportion factor; *b(t)* is the learning rate which is chosen satisfy the two conditions: As $t \to \infty$; $b(t) \to 0$ and $b(t) \to \infty$. *b(t)* can be taken as

$$b(t) = b(t-1) / \sqrt{1 + t/n} \tag{9}$$

Unlike the n-means clustering algorithm, where only the winning prototype is updated, the GLVQ updates all nodes for a given input vector. The learning rules depend on the degree of the distance match to the winner node; the lesser the degree of match with the winner, the more is the impact on nonwinner nodes. The GLVQ is more robust to initialize. Furthermore, the algorithm can give more reasonable solution when *D>1*. But it has been found that GLVQ behaves exactly opposite to what was desired when *D<1*.

To overcome this problem, we can give a fuzzy modification of GLVQ. Let L_x be a loss function which measures the locally weighted mismatch (error) with respect to the winner

$$L_x = L(x, c_1, c_2, \ldots, c_n) = \sum_{j=1}^{n} \mu_j \left\|x - c_j\right\|^2 \tag{10}$$

where μ_j is chosen to be a fuzzy member function as

$$\mu_j = \left(\sum_{p=1}^{n} \frac{\left\|x - c_j\right\|^2}{\left\|x - c_p\right\|^2} \right)^{-1} \tag{11}$$

which has the following properties

1. The magnitude of u_j is inversely proportional $\left|x - c_j\right|$;
2. Each j is within *[0,1];*
3. The sum of j are equal to *1.*

Assume $\left\|x - c_p\right\| > 0$, the gradient of L_x with respect to c_p can be calculated as

$$\nabla_{c_p} L = -2n\mu_p^2 (x - c_p) \tag{12}$$

Thus, the updating equation of the kernels can be rewritten as

$$c_j(t+1) = c_j(t) + 2b(t)n\mu_j^2 \left(x(t) - c_j(t)\right) \tag{13}$$

3.1.2 Adaptive Modification of FGLVQ

In adaptive identification and control, since a plant is unknown and the structure of a given TSWNN is limited, therefore the modeling error exists inevitably. In most case, the TSWNN with fixed kernels can not universally approximate the plant in the whole domain with the uniform error accuracy. So the kernels and the weights should be persistently adjusted to trace the output error simultaneously. Here we give an updating equation of *b(t)*

$$b(t) = b_o \left\|\varepsilon(t)\right\| / \left(1 + \left\|\varepsilon(t)\right\|\right) \tag{14}$$

where $0 < b_o < 1$, thus $0 < b(t) < 1$. *b(t)* varies with $\varepsilon(t)$. Of course, when $\varepsilon(t)$ converges to zero, *b(t)* will tend to zero too.

3.2 Recursive Least Squares with Variable Forgetting Factor

Define the hidden layer output vector at the instant *t* as

$$\Phi_t = [\phi_1(t), \phi_2(t), \ldots, \phi_n(t)]^T \tag{15}$$

The connection weight vector at *t* as

$$\Theta_t = [\theta_1(t), \theta_2(t), \ldots, \theta_n(t)]^T \tag{16}$$

Thus, Θ_t can be recursively updated by the equations

$$\Theta_{t+1} = \Theta_t + K_t \varepsilon(t) \tag{17}$$

$$K_t = \frac{P_t \Phi_t}{1 + \Phi_t^T P \Phi_t} \tag{18}$$

$$P_{t+1} = (I - K_t \Phi_t^T) P_t / \rho_t \tag{19}$$

$$\rho_t = 1 - (1 + \Phi_t^T K_t) \varepsilon_t^2 / \sum{}_o \tag{20}$$

If $\rho_t < \rho_{\min}$, set $\rho_t = \rho_{\min}$; $\sum{}_o > 0$ reflects the amplitude of the noise.

4 Adaptive Control Law Optimization

The following gives the indirect TSWNN-based adaptive controller. On the basis of the one-step ahead prediction TSWNN model, we use a numerically stable Davidon's least squares-based minimization approach to optimize the control law iteratively in each sample period. Define the predictive control error

$$e(t+1)=\hat{y}(t+1)-\left[\alpha y(t)+(1-\alpha)y_r(t+1)\right] \tag{21}$$

where $y_r(t+1)$ is the reference output signal; $0<\alpha<1$ is a smoothing factor.
The goal of iteratively optimizing control law is to make *e(t+1)* small enough to reach some control accuracy. Let the initial value of *u(t)* be

$$\mu_o(t)=\mu(t-1) \tag{22}$$

and the corresponding output of the TSWNN be

$$\hat{y}_o(t+1)=f_r(x_o(t+1)) \tag{23}$$

where

$$x_o(t+1)=[y(t),...,y(t-n_y+1);u(t),...,u(t-n_u+1)] \tag{24}$$

Similarly, $u_k(t)$ denotes the control value at the *k* iteration instant. The corresponding output of the TSWNN is

$$\hat{y}_k(t+1)=f_r(x_k(t+1)) \tag{25}$$

where

$$x_k(t+1)=[y(t),...,y(t-n_y+1);u_k(t),...,u_k(t-n_u+1)] \tag{26}$$

and

$$e_k(t+1)=\hat{y}(t+1)-\left[\alpha y(t)+(1-\alpha)y_r(t+1)\right] \tag{27}$$

The optimization procedure is described as follows.

* Let $k_{\max}$ be the maximum iteration number and $e_{\max}$ be the desired control accuracy. Set the initial Hessian matrix $H_o=h_oI, 0<h_o<1, k=0$.

* Let *k=k+1. u(t)* at the *k* iteration is calculated by

$$\mu_k(t)=\mu_{k-1}(t)-H_k\nabla\hat{y}_{k-1}e_{k-1} \tag{28}$$

$$H_k=(\lambda_c H_{k-1}^{-1}+\nabla^T\hat{y}_{k-1}+\alpha_c I)^{-1} \tag{29}$$

where $0<\lambda_c<1, 0<\alpha_c<1$; $\nabla\hat{y}_{k-1}$ is the gradient of $y_{k-1}(t+1)$ with respect to $u_{k-1}(t)$.

* If $k = k_{\max}$ or $e_k(t+1) = e_{\max}$, then stop calculating; Otherwise, go to (2). In the DLS algorithm, the updating equation for H_k is

$$H_k = (\lambda_c H_{k-1}^{-1} + \nabla \hat{y}_{k-1} \nabla^T \hat{y}_{k-1})^{-1} \tag{30}$$

The item $\alpha_c I$ of the equation (29) is not included in the equation (30). It has been found that when $\hat{y}_{k-1}$ is small and $0 < \lambda_c < 10$, the H_k will become divergent infinitely as k increases. This can result in numeric instability. In the equation (29), the finite boundary of H_k is guaranteed.

5 Simulation Results

Consider the first-order non-linear system described by:

$$y(k) = \frac{y(k-10)}{1+y^2(k-1)} + u^3(k-1) \tag{31}$$

The parameters for the proposed algorithm are chosen as

$m = 7, h_o = 0.09, b_o = 0.21, \lambda_c = 0.001, \varepsilon(t) = e^{-800t}$

Simulation results are shown in figures 1 and 2. In figure 1 the problem of set point tracking of the closed-loop system is shown. Figure 2 indicates growth pattern (number of rules). As it can be seen, the performance of the closed-loop system is excellent.

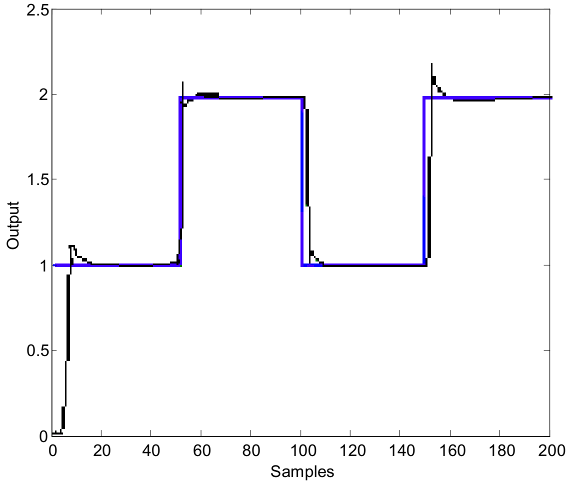


Fig. 1. Set point tracking; Set point trajectory and output.

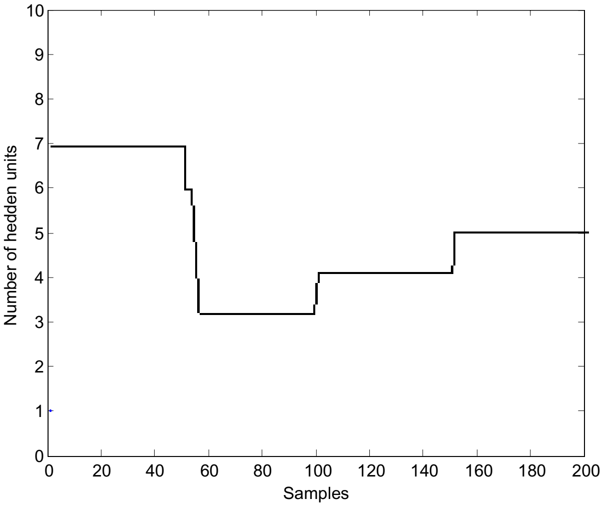


Fig. 2. Growth pattern (number of rules).

6 Conclusion

In this paper, firstly, we proposed a novel approach to train the TSWNN. An adaptive fuzzy generalized learning vector quantization (AFGLVQ) technique was adopted to adjust the kernels of the TSWNN, and recursive least squares with variable forgetting factor (VRLS) was applied to update the connection weights of the network. Secondly, on the basis of the one-step ahead TSWNN predictor, a numerically stable Davidon's least squares-based minimization approach was used to optimize the control law iteratively in each sample period. The simulations demonstrated the rapid learning and adaptive property of the identification algorithm and the effectiveness of control algorithm. The scheme is especially suitable for controlling the SISO nonlinear systems.

References:

1. Wang Shoujue. The Sequential Learning Ahead Masking (SLAM) model of neural networks for pattern classification. *Proceedings of JCIS98,* pp.199-202. Oct.. RTP. NC. USA, 1998.
2. Nelles, O. Nonlinear system identification: from classical approach to neuro-fuzzy identification. Springer-Verlag, 2001.
3. Wang Shoujue. Direction-Basis-Function neural networks, *Proceedings of IJCNN'99*, pp.1251-2171, July, Washington DC, USA, 1999.
4. Cao S.G., Rees N.W., Feng G. Analysis and design for a complex control system, part I: fuzzy modeling and identification. *Automatic*a, 6(8), pp.1017-1028, 1997.

5. Yingwei Lu, N .Sundarajan and P. Saratchandran, Identification of time-varying nonlinear systems using minimal radial basis function neural networks, *IEE Proceedings: Control Theory and Applications*. 144(2), pp. 202-208, 1997.
6. Fabri S., and Kadirkamanatham V., Dynamic Structure Neural Networks for Stable Adaptive Control of Nonlinear Systems. *IEEE Transaction on Neural Networks*, 7(5), pp.1151-1167, 1996.
7. Jeffrey T.S. and Kevin M. Passino, Stable Adaptive Control Using Fuzzy Systems and Neural Networks, IEEE Transactions on Fuzzy Systems 4(3), pp.339-359, 1996.
8. Sanner R.M. and Jean-Jacques E. Slotine, Gaussian Networks for Direct Adaptive Control, *IEEE Transaction on Neural Networks*, 3(6), pp.837-867, 1992.

Knowledge Discovery Using Neural Networks

Khosrow Kaikhah, and Sandesh Doddameti

Department of Computer Science
Texas State University
San Marcos, Texas 78666
{kk02, sd101017}@TxState.edu

Abstract. A novel knowledge discovery technique using neural networks is presented. A neural network is trained to learn the correlations and relationships that exist in a dataset. The neural network is then pruned and modified to generalize the correlations and relationships. Finally, the neural network is used as a tool to discover all existing hidden trends in four different types of crimes in US cities as well as to predict trends based on existing knowledge inherent in the network.

1 Introduction

Enormous amounts of data are being generated and recorded for almost any kind of event or transaction. Advances in data storage and database technology have enabled us to store the vast amount of data very efficiently. A small piece of data may be quite insignificant. However, taken as a whole, data encompasses a vast amount of knowledge. A vital type of knowledge that can be acquired from large datasets are the hidden trends. These hidden trends, which can be expressed as rules or correlations, highlight the associations that exist in the data. For example, in a financial institution environment, where information about customer's characteristics and activities are maintained, the following trend may exist.

Persons who are between 25-30 years old, having at least a bachelor's degree with an income greater than 50K, have greater than 6 entertainment activities and greater than 10 restaurant activities in each cycle.

Finding these trends, which are specific to the application, represent a type of knowledge discovery. The acquired knowledge is helpful in understanding the domain, which the data describes.

We define a machine learning process that uses artificial neural networks to discover trends in large datasets. A neural network is trained to learn the inherent relationships among the data. The neural network is then modified via pruning and hidden layer activation clustering. The modified neural network is then used as a tool to extract common trends that exist in the dataset as well as to predict trends. The extraction phase can be regulated through several control parameters.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 20-28, 2004.

2 Related Research

Andrews et al. in [1] discuss the difficulty in comprehending the internal process of how a neural network learns a hypothesis. According to their survey, rule extraction methods have been categorized into decompositional and pedagogical techniques. They discuss various techniques including the pros and cons of each method. The distinguishing characteristic of the decompositional approach is that the focus is on extracting rules at the level of individual (hidden and output) units within the trained Artificial Neural Network. In pedagogical approach to rule extraction, the trained neural network is treated as a black-box, in other words, the view of the underlying trained artificial neural network is opaque. They conclude that no single rule extraction/rule refinement technique is currently in a dominant position to the exclusion of all others.

Gupta et al. in [2] propose an algorithm (GLARE) to extract classification rules from feedforward and fully connected neural networks trained by backpropagation. The major characteristics of the GLARE algorithm are (a) its analytic approach for rule extraction, (b) its applicability to standard network structure and training method, and (c) its rule extraction mechanism as direct mapping between input and output neurons. This method is designed for a neural network with only one hidden layer. This approach uses the significance of connection strengths based on their absolute magnitude and uses only a few important connections (highest absolute values) to analyze the rules.

Our knowledge discovery process is both decompositional and pedagogical. It is decompositional in nature, since we examine the weights for pruning and clustering the hidden unit activation values. It is pedagogical, since we use the neural network as a black-box for knowledge discovery. Our approach is neither limited by the complexity of the hidden layer, nor by the number of hidden layers. Therefore our approach can be extended to networks with several hidden layers.

3 Our Approach

We have developed a novel process for discovering knowledge in datasets, with *m* dimensional input space and *n* dimensional output space, utilizing neural networks. Our process is independent of the application. The significance of our approach lies in using neural networks for discovering knowledge, with control parameters. The control parameters influence the discovery process in terms of importance and significance of the acquired knowledge. There are four phases in our approach: 1) neural network training, 2) pruning and re-training, 3) clustering the hidden neuron activation values, and 4) rule discovery and extraction.

In phase one, the neural network is trained using a supervised learning method. The neural network learns the associations inherent in the dataset. In phase two, the neural network is pruned by removing all unnecessary connections and neurons. In phase three, the activation values of the hidden layer neurons are clustered using an adaptable clustering technique. In phase four, the modified neural network is used as a tool to extract and discover hidden trends. These four phases are described in more detail in the next four sections.

3.1 Neural Network Training

Neural networks are able to solve highly complex problems due to the non-linear processing capabilities of their neurons. In addition, the inherent modularity of the neural network's structure makes them adaptable to a wide range of applications [3]. The neural network adjusts its parameters to accurately model the distribution of the provided dataset. Therefore, exploring the use of neural networks for discovering correlations and trends in data is prudent.

The input and output patterns may be real-valued or binary-valued. If the patterns are real-valued, each value is discretized and represented as a sequence of binary values, where each binary value represents a range of real values. For example, in a credit card transaction application, an attribute may represent the person's age (a value greater than 21). This value can be discretized into 4 different intervals: (21-30],(30-45],(45-65], and (65+]. Therefore [0 1 0 0] would represent a customer between the ages of 31 and 45. The number of neurons in the input and output layers are determined by the application, while the number of neurons in the hidden layer are dependent on the number of neurons in the input and output layers.

We use an augmented gradient descent approach to train and update the connection strengths of the neural network. The gradient descent approach is an intelligent search for the global minima of the energy function. We use an energy function, which is a combination of an error function and a penalty function [4]. The error function computes the error of each neuron in the output layer, and the penalty function drives the connection strengths of unnecessary connections to very small values while strengthening the rest of the connections. The penalty function is defined as:

$$P(w,v) = \rho_{decay}\left(P_1(w,v) + P_2(w,v)\right) \tag{1}$$

$$P_1(w,v) = \varepsilon_1\left(\sum_{j=1}^{h}\sum_{i=1}^{m}\frac{\beta w_{ij}^2}{1+\beta w_{ij}^2} + \sum_{j=1}^{h}\sum_{k=1}^{n}\frac{\beta v_{jk}^2}{1+\beta v_{jk}^2}\right) \tag{1a}$$

$$P_2(w,v) = \varepsilon_2\left(\sum_{j=1}^{h}\sum_{i=1}^{m} w_{ij}^2 + \sum_{j=1}^{h}\sum_{k=1}^{n} v_{jk}^2\right) \tag{1b}$$

The network is trained till it reaches a recall accuracy of 99% or higher.

3.2 Pruning and Re-training

The neural network is trained with an energy function, which includes a penalty function. The penalty function drives the strengths of unnecessary connections to approach zero very quickly. Therefore, the connections having very small values, values less than 1, can safely be removed without significant impact on the performance of the network. After removing all weak connections, any input layer neuron having no emanating connections can be removed. In addition, any hidden layer neuron having no abutting or emanating connections can safely be removed. Finally, any output layer neuron having no abutting connections can be removed. Removal of input layer neurons correspond to having irrelevant inputs in the data model; removal of hidden layer neurons reduces the complexity of the network and the clustering phase; and removal of the output layer neurons corresponds to having irrelevant outputs in the data model. Pruning the neural network results in a less complex network while improving its generalization.

Once the pruning step is complete, the network is trained with the same dataset in phase one to ensure that the recall accuracy of the network has not diminished significantly. If the recall accuracy of the network drops by more than 2%, the pruned connections and neurons are restored and a stepwise approach is pursued. In the stepwise pruning approach, the weak incoming and outgoing connections of the hidden layer neurons are pruned, one neuron at a time, and the network is re-trained and tested for recall accuracy.

3.3 Clustering the Hidden Layer Neuron Activation Values

The activation values of each hidden layer neuron are dynamically clustered and re-clustered with a cluster radius and confidence radius, respectively. The clustering algorithm is adaptable, that is, the clusters are created dynamically as activation values are added into the clusterspace. Therefore, the number of clusters and the number of activation values in each cluster are not known *a priori*. The centroid of each cluster represents the mean of the activation values in the cluster and can be used as the representative value of the cluster, while the frequency of each cluster represents the number of activation values in that cluster. By using the centroids of the clusters, each hidden layer neuron has a minimal set of activations. This helps with getting generalized outputs at the output layer. The centroid of a cluster c is denoted by G_c. The centroid is adjusted dynamically as new elements e_c^i are added to the cluster.

$$G_c^{new} = \frac{(G_c^{old} \cdot freq_c) + e_c^i}{freq_c + 1} \tag{2}$$

Since dynamic clustering is order sensitive, once the clusters are dynamically created with a cluster radius that is less than a predetermined upper bound, all elements will be re-clustered with a confidence radius of one-half the cluster radius. The upper bound for cluster radius defines a range for which the hidden layer neuron activation values can fluctuate without compromising the network performance.

The benefits of re-clustering are twofold: 1) Due to order sensitivity of dynamic clustering, some of the activation values may be misclassified. Re-clustering alleviates this deficiency by classifying the activation values in appropriate clusters. 2) Re-clustering with a different radius (confidence radius) eliminates any possible overlaps among clusters. In addition during re-clustering, the frequency of each confidence cluster is calculated, which will be utilized in the extraction phase.

3.4 Knowledge Discovery

In the final phase of the process, the knowledge acquired by the trained and modified neural network is extracted in the form of rules [5], [6]. This is done by utilizing the generalization of the hidden layer neuron activation values as well as control parameters. The novelty of the extraction process is the use of the hidden layer as a filter by performing vigilant tests on the clusters. Clusters identify common regions of activations along with the frequency of such activities. In addition, clusters provide representative values (the mean of the clusters) that can be used to retrieve generalized outputs.

The control parameters for the extraction process include: a) cluster radius, b) confidence frequency, and c) hidden layer activation level. The cluster radius determines the coarseness of the clusters. The confidence radius is usually set to one-half of the cluster radius to remove any possible overlaps among clusters. The confidence frequency defines the minimum acceptable rate of commonality among patterns. The hidden layer activation level defines the maximum level of tolerance for inactive hidden layer neurons.

Knowledge extraction is performed in two steps. First, the existing trends are discovered by presenting the input patterns in the dataset to the trained and modified neural network and by providing the desired control parameters. The input patterns that satisfy the rigorous extraction phase requirements and produce an output pattern represent generalization and correlations that exist in the dataset. The level of generalization and correlation acceptance is regulated by the control parameters. This ensures that inconsistent patterns, which fall outside confidence regions of hidden layer activations, or fall within regions with low levels of activity, are not considered. There may be many duplicates in these accepted input-output pairs. In addition, several input-output pairs may have the same input pattern or the same output pattern. Those pairs having the same input patterns will be combined, and, those pairs having the same output patterns will be combined. This post-processing is necessary to determine the minimal set of trends. Any input or output attribute not included in the discovered trend corresponds to irrelevant attributes in the dataset. Second, the predicated trends are extracted by providing all possible permutations of input patterns, as well as the desired control parameters. Any additional trends discovered in this step constitute the predicated knowledge based on existing knowledge. This step is a direct byproduct of the generalizability of neural networks.

4 Discovering Trends in Crimes in US Cities

We compiled a dataset consisting of the latest annual demographic and crime statistics for 6100 US cities. The data is derived from three different sources: 1) US Census; 2) Uniform Crime Reports (UCR) published annually by the Federal Bureau of Investigation (FBI); 3) Unemployment Information from the Bureau of Labor Statistics.

We used the dataset to discover trends in crimes with respect to the demographic characteristics of the cities. We divided the dataset into three groups in terms of the population of cities: a) cities with populations of less than 20k (4706 cities), b) cities with populations of greater than 20k and less than 100k (1193 cities), and c) cities with populations of greater than 100k (201 cities). We then trained a neural network for each group and each of four types of crimes (murder, rape, robbery, and auto theft), a total of 12 networks. We divided the dataset into three groups in terms of city population, since otherwise, small cities (cities less that 20k) would dominate the process due to their high overall percentage. Table 1 includes the demographic characteristics and crime types we used for the knowledge discovery process.

Table 1. The Categories for the Process

I_1: *City Population*
I_2: *Percentage of Single-Parent Households*
I_3: *Percentage of Minority*
I_4: *Percentage of Young People(between the ages of 15 and 24)*
I_5: *Percentage of Home Owners*
I_6: *Percentage of People living in the Same House since 1985*
I_7: *Percentage of Unemployment*

O_1: *Number of Murders*
O_2: *Number of Rapes*
O_3: *Number of Robberies*
O_4: *Number of Auto Thefts*

Table 2. Discrete Intervals

Categories	N	Intervals
I_1 (*small*)	5	[0-4k],(4k-8k],(8k,12k],(12k-16k],(16k-20]
I_1 (*medium*)	5	(20k-40k],(40k-60k],(60k-80k],(80k-90k],(90k-100k]
I_1 (*large*)	5	(100k-130k],(130k-160k],(160k-200k],(200k-500k],500k+
I_2	7	[0-5],(5-7],(7-9],(9-11],(11-14],(14-20],(20-100]
I_3	6	[0-5],(5-10],(10-20],(20-40],(40-70],(70-100]
I_4	7	[0-12],(12-13],(13-14],(14-15],(15-17],(17-25],(25-100]
I_5	7	[0-40],(40-50],(50-60],(60-70],(70-80],(80-90],(90-100]
I_6	6	[0-45],(45-50],(50-55],(55-60],(60-65],(65-100]
I_7	6	[0-4],(4-6],(6-8],(8-12],(12-20],(20-100]
O_1	4	0, (1-5],(5-10],10+
O_2	5	0, (1-5],(5-10],(10-70],70+
O_3	5	0, (1-5],(5-10],(10-100],100+
O_4	5	[0-10],(10-100],(100-500],(500-1000],1000+

Each category is discretized into several intervals to define the binary input/output patterns. For each crime type, three different neural networks are trained for the three groups of cities (small, medium, and large) to an accuracy of 99% or higher. Each network consists of 44 input layer neurons, 60 hidden layer neurons, and 4 to 5 output layer neurons. After the training phase, the networks are pruned and clustered. Although, for each network, about 30% of connections as well as about 5% of hidden layer neurons were pruned, none of the input neurons were pruned. This reflects the importance of all demographic categories we used for discovering trends in crimes. After phase two and three, all networks maintain an accuracy rate of 99% or higher. The networks were then used as tools to discover the existing, as well as predicted trends. Table 2 represents the discrete intervals for each category.

4.1 Trends in Small Cities

The following are the existing trends discovered for small cities.

1) $(0<I_1 \le 4k)\wedge(0<I_3 \le 5)\wedge(0<I_7 \le 4)\wedge(7<I_2 \le 9)\wedge(50<I_6 \le 55)\wedge(14<I_4 \le 15)\wedge$
$(70<I_5 \le 80)\Rightarrow O_1 = 0$

2) $(0k<I_1 \le 8k)\wedge(0<I_3 \le 5)\wedge(0<I_7 \le 4)\wedge[(0<I_2 \le 5)\vee(7<I_2 \le 9)]\wedge(0<I_6 \le 50)\wedge$
$(0<I_4 \le 13)\wedge(60<I_5 \le 70)\Rightarrow 1<O_3 \le 5$

3) $(4k<I_1 \le 8k)\wedge(0<I_3 \le 5)\wedge(6<I_7 \le 8)\wedge(9<I_2 \le 11)\wedge(65<I_6 \le 80)\wedge$
$(13<I_4 \le 14)\wedge(60<I_5 \le 70)\Rightarrow 1<O_4 \le 5$

4) $(8k<I_1 \le 16k)\wedge(0<I_3 \le 5)\wedge[(0<I_7 \le 4)\vee(8<I_7 \le 12)]\wedge(7<I_2 \le 9)\wedge$
$(55<I_6 \le 60)\wedge(13<I_4 \le 14)\wedge(60<I_5 \le 70)\Rightarrow O_4 = 0$

The following are the predicted trends discovered for small cities.

1) $[(0<I_1 \le 4k)\vee(12k<I_1 \le 16k)]\wedge(0<I_3 \le 5)\wedge(0<I_7 \le 6)\wedge(0<I_2 \le 5)\wedge$
$(0<I_6 \le 55)\wedge(13<I_4 \le 14)\wedge(70<I_5 \le 80)\Rightarrow O_1 = 0$

2) $(0<I_1 \le 4k)\wedge(20<I_3 \le 40)\wedge(4<I_7 \le 6)\wedge(9<I_2 \le 11)\wedge(0<I_6 \le 45)\wedge$
$(13<I_4 \le 14)\wedge(50<I_5 \le 60)\Rightarrow O_2 = 0$

3) $(4k<I_1 \le 8k)\wedge(5<I_3 \le 10)\wedge(4<I_7 \le 6)\wedge(5<I_2 \le 7)\wedge(60<I_6 \le 65)\wedge$
$(15<I_4 \le 17)\wedge(40<I_5 \le 50)\Rightarrow 1<O_2 \le 5$

4) $(12k<I_1 \le 16k)\wedge(5<I_3 \le 10)\wedge(4<I_7 \le 6)\wedge(5<I_2 \le 7)\wedge(0<I_6 \le 45)\wedge$
$(17<I_4 \le 25)\wedge(80<I_5 \le 90)\Rightarrow 1<O_3 \le 5$

4.2 Trends in Medium Cities

The following are the existing trends discovered for medium cities.

1) $(20k<I_1 \le 40k)\wedge(0<I_3 \le 5)\wedge(0<I_7 \le 6)\wedge[(5<I_2 \le 7)\vee(9<I_2 \le 11)]\wedge$
$(45<I_6 \le 55)\wedge(12<I_4 \le 13)\wedge(60<I_5 \le 90)\Rightarrow O_1 = 0$

2) $(20k<I_1 \le 40k)\wedge(10<I_3 \le 20)\wedge(0<I_7 \le 4)\wedge(5<I_2 \le 7)\wedge$
$(0<I_6 \le 45)\wedge(12<I_4 \le 13)\wedge(60<I_5 \le 70)\Rightarrow 1<O_2 \le 5$

3) $(20k<I_1 \le 40k)\wedge(5<I_3 \le 10)\wedge(4<I_7 \le 6)\wedge(11<I_2 \le 14)\wedge$
$(45<I_6 \le 50)\wedge(14<I_4 \le 15)\wedge(40<I_5 \le 50)\Rightarrow 1<O_2 \le 10$

4) $(20k<I_1 \le 40k)\wedge(0<I_3 \le 5)\wedge(4<I_7 \le 6)\wedge(0<I_2 \le 5)\wedge$
$(65<I_6 \le 80)\wedge(12<I_4 \le 13)\wedge(80<I_5 \le 90)\Rightarrow 10<O_4 \le 100$

5)
$$\begin{aligned}&(20k < I_1 \leq 40k)\wedge(0 < I_3 \leq 5)\wedge(4 < I_7 \leq 6)\wedge(9 < I_2 \leq 11)\wedge\\&(0 < I_6 \leq 45)\wedge(14 < I_4 \leq 15)\wedge(50 < I_5 \leq 60)\Rightarrow 100 < O_4 \leq 500\end{aligned}$$

The following are the predicted trends discovered for medium cities.

1)
$$\begin{aligned}&(20 < I_1 \leq 40k)\wedge(40 < I_3 \leq 100)\wedge[(0 < I_7 \leq 4)\vee(6 < I_7 \leq 8)]\wedge(5 < I_2 \leq 9)\wedge\\&(40 < I_6 \leq 55)\wedge[(13 < I_4 \leq 14)\vee(17 < I_4 \leq 25)]\wedge(40 < I_5 \leq 60)\Rightarrow 1 < O_2 \leq 10\end{aligned}$$

2)
$$\begin{aligned}&(60k < I_1 \leq 80k)\wedge(10 < I_3 \leq 20)\wedge(4 < I_7 \leq 6)\wedge(7 < I_2 \leq 9)\wedge\\&(50 < I_6 \leq 55)\wedge(15 < I_4 \leq 17)\wedge(80 < I_5 \leq 90)\Rightarrow O_3 = 0\end{aligned}$$

3)
$$\begin{aligned}&(20k < I_1 \leq 40k)\wedge(70 < I_3 \leq 100)\wedge(6 < I_7 \leq 8)\wedge(5 < I_2 \leq 7)\wedge\\&(0 < I_6 \leq 45)\wedge(0 < I_4 \leq 12)\wedge(80 < I_5 \leq 90)\Rightarrow 10 < O_3 \leq 100\end{aligned}$$

4.3 Trends in Large Cities

The following are the existing trends discovered for large cities.

1)
$$\begin{aligned}&(200k < I_1 \leq 500k)\wedge(20 < I_3 \leq 40)\wedge(8 < I_7 \leq 12)\wedge(11 < I_2 \leq 14)\wedge\\&(50 < I_6 \leq 55)\wedge(15 < I_4 \leq 17)\wedge(50 < I_5 \leq 60)\Rightarrow 5 < O_1 \leq 10\end{aligned}$$

2)
$$\begin{aligned}&(200k < I_1 \leq 500k)\wedge(40 < I_3 \leq 70)\wedge(8 < I_7 \leq 12)\wedge(14 < I_2 \leq 20)\wedge\\&(50 < I_6 \leq 55)\wedge(15 < I_4 \leq 17)\wedge(50 < I_5 \leq 60)\Rightarrow 10 < O_2 \leq 70\end{aligned}$$

3)
$$\begin{aligned}&(160k < I_1 \leq 200k)\wedge(20 < I_3 \leq 40)\wedge(6 < I_7 \leq 8)\wedge(11 < I_2 \leq 14)\wedge\\&(45 < I_6 \leq 50)\wedge(14 < I_4 \leq 17)\wedge(50 < I_5 \leq 60)\Rightarrow O_3 > 100\end{aligned}$$

4)
$$\begin{aligned}&(100k < I_1 \leq 130k)\wedge(10 < I_3 \leq 20)\wedge(0 < I_7 \leq 4)\wedge(5 < I_2 \leq 7)\wedge\\&(0 < I_6 \leq 45)\wedge(0 < I_4 \leq 12)\wedge(70 < I_5 \leq 80)\Rightarrow 500 < O_4 \leq 1000\end{aligned}$$

The following are the predicted trends discovered for large cities.

1)
$$\begin{aligned}&(200k < I_1 \leq 500k)\wedge(70 < I_3 \leq 100)\wedge(6 < I_7 \leq 8)\wedge(11 < I_2 \leq 14)\wedge\\&[(0 < I_6 \leq 45)\vee(50 < I_6 \leq 55)]\wedge[(13 < I_4 \leq 14)\vee(17 < I_4 \leq 25)]\Rightarrow O_2 > 70\end{aligned}$$

2)
$$\begin{aligned}&(100k < I_1 \leq 130k)\wedge(70 < I_3 \leq 100)\wedge(6 < I_7 \leq 8)\wedge(11 < I_2 \leq 14)\wedge\\&[(45 < I_6 \leq 50)\vee(60 < I_6 \leq 65)]\wedge(17 < I_4 \leq 25)\wedge(50 < I_5 \leq 60)\Rightarrow O_3 > 10(\end{aligned}$$

3)
$$\begin{aligned}&(200k < I_1 \leq 500k)\wedge(70 < I_3 \leq 100)\wedge(8 < I_7 \leq 12)\wedge(11 < I_2 \leq 14)\wedge\\&(60 < I_6 \leq 65)\wedge(0 < I_4 \leq 12)\wedge(60 < I_5 \leq 70)\Rightarrow O_4 > 1000\end{aligned}$$

4)
$$\begin{aligned}&(100k < I_1 \leq 130k)\wedge(5 < I_3 \leq 10)\wedge(12 < I_7 \leq 20)\wedge(9 < I_2 \leq 11)\wedge\\&(45 < I_6 \leq 50)\wedge(13 < I_4 \leq 14)\wedge(0 < I_5 \leq 40)\Rightarrow 100 < O_4 \leq 1000\end{aligned}$$

5 Conclusions

For each group of cities (small, medium, large), we are able to discover the existing trends for each type of crime (murder, rape, robbery, auto theft). These trends represent the hidden knowledge and are based on the high level of commonalty inherent in the dataset. The desired level of commonalty can be regulated through the control parameters. In addition, by using the generalizability feature of neural networks, we are able to discover predicted trends. These trends describe the demographic characteristics of cities that contribute to each type of crime. Once again, the control parameters provide the ability to regulate the desired level of commonality. According to the experts in criminal fields, the discovered trends accurately reflect the reality that exists in US cities. They were particularly impressed with the predicted trends, since they can use this knowledge for restructuring their resources.The knowledge discovery technique can be applied to any application domain that deals with vast amounts of data such as medical, military, business, and security. In medical fields, the data gathered from cancer patients can be used to discover the dominating factors and trends for the development of cancer. In military fields, the data gathered from the enemy can be used to predicate their future movements. In business environments, the data gathered from customers can be used to model the transaction activities of the customers. In security applications, the data gathered can be used to predicate and prevent potential intrusions.

References

1. Robert Andrews, Joachim Diederich, and Alan Tickle, “A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks”, Neurocomputing Research Center, 1995
2. Amit Gupta, Sang Park, and Siuva M. Lam, “Generalized Analytic Rule Extraction for Feedforward Neural Networks”, IEEE transactions on knowledge and data engineering, 1999
3. Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka, Elements of Artificial Neural Networks (Complex Adaptive Systems), Cambridge, MA: MIT Press, 1997
4. Rudy Setiono, “Extracting Rules from Pruned Neural Networks for Breast Cancer Diagnosis”, Artificial Intelligence in Medicine, 1996
5. Rudy Setiono and Huan Liu, “Effective Data Mining Using Neural Networks”, *IEEE transactions on knowledge and data engineering,* 1996
6. Tony Kai, Yun Chan, Eng Chong Tan, and Neeraj Haralalka, “A Novel Neural Network for Data Mining”, *8th International Conference on Neural Information Processing Proceedings. Vol.2*, 2001
7. Mark W. Craven and Jude W. Shavlik, “Using Neural Networks for Data Mining”, *Future Generation Computer Systems special issue on Data Mining,* 1998
8. Jason T. L. Wang, Qicheng Ma, Dennis Shasha, and Cathy H.Wu, kkk“Application of Neural Networks to Biological Data Mining: A Case Study in Protein Sequence Classification”, *The Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* August 20-23, 2000 Boston, MA, USA.

Knowledge Discovery in Hepatitis C Virus Transgenic Mice

A. Fazel Famili[1], Junjun Ouyang[1], Marko Kryworuchko[2], Ikuri Alvarez-Maya[2], Brandon Smith[3], and Francisco Diaz-Mitoma[2]

[1] Institute for Information Technology, Bldg. M-50 NRC, Ottawa, On. K1A0R6 Canada
{fazel.famili, junjun.ouyang}@nrc.gc.ca

[2] Children's Hospital of Eastern Ontario, 401 Smyth Rd. Ottawa, On. K1H 8L1 Canada
{MKryworuchko, IAlvarez-Maya, diaz}@cheo.on.ca

[3] Institute for Biological Sciences, Bldg. M-54 NRC, Ottawa, On. K1A0R6 Canada
{brandon.smith}@nrc.gc.ca

Abstract. For the purpose of gene identification, we propose an approach to gene expression data mining that uses a combination of unsupervised and supervised learning techniques to search for useful patterns in the data. The approach involves validation and elimination of irrelevant data, extensive data pre-processing, data visualization, exploratory clustering, pattern recognition and model summarization. We have evaluated our method using data from microarray experiments in a Hepatitis C Virus transgenic mouse model. We demonstrate that from a total of 15311 genes (attributes) we can generate simple models and identify a small number of genes that can be used for future classifications. The approach has potential for future disease classification, diagnostic and virology applications.

1 Introduction

The field of bioinformatics involves a close link with a number of diverse research areas, from genomics and proteomics to computer science, mathematics and in particular data mining. This collaboration of disciplines has evolved because of: (i) the advances in data production and acquisition facilities, such as mircroarrays and high throughput genomics, (ii) the enormous amounts of data that cannot be analyzed using ordinary tools, and (iii) the strong interest from many groups (research institutes, hospitals, pharmaceuticals, etc.) who want to benefit from this wealth of data. Advancements in microarray technology, for example, have overwhelmed scientists with expression profiles of tens of thousands of genes from a variety of organisms. Researchers have undertaken many efforts to deal with these issues [5, 6, 8, 13, 15 and 16], and have noticed the lack of powerful and efficient knowledge discovery tools, along with well defined knowledge discovery strategies.

Knowledge discovery is the process of developing strategies to discover ideally all previously unknown knowledge from historical or real time data. Microarray related applications expect that the knowledge discovery process will help, such that one can (i) identify anomalies of certain genes or experiments, (ii) define relationships

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 29–39, 2004.

between genes and their functions based on expression profiles, and (iii) build diagnosis models for clinical classes, e.g. normal and diseased organs.

With expression profiles from thousands of genes, the specific objectives of this study were (i) to develop a data mining strategy that can deal with a relatively large amount of microarray data, (ii) to efficiently search for patterns and meaningful relations and (iii) to identify genes that can differentiate between mice expressing Hepatitis C Virus proteins (transgenic) and non-expressing age matched controls (non-transgenic). We first explain the research problem in section 2. In section 3, we describe the data collection process and briefly introduce BioMiner software. In section 4 we give an overview of the data preprocessing. Section 5 describes our knowledge discovery process and section 6 presents the results. We conclude the paper in section 7.

2 The Research Problem

Hepatitis C virus (HCV) constitutes a major cause of chronic liver disease around the world. Approximately 200 million people worldwide are infected with HCV [4, 10, and 15]. The development of a suitable vaccine against HCV is a complicated and difficult task due to the broad genetic variability of the virus genome allowing it to escape control by the host immune response. There have been genomic studies on HCV using various models [1, 10, 11, 12, 13 and 15]. The lack of good *in vitro* models as well as small animal models of infection have hampered medical researchers' abilities to characterize the mechanism by which the virus causes liver damage and to identify correlates of protection.

In this study, transgenic mice expressing HCV core (E1 and E2) proteins were produced to exhibit liver abnormalities similar to those of natural HCV infections. Researchers are interested to compare gene expression in the livers of HCV-transgenic mice to that of non-transgenic mice and correlate this with the pathology. We obtained gene expression data from HCV-transgenic experiments. We then applied our data processing and analysis tools and related data mining technology to examine the data, look into possible anomalies, build explicit models, and identify important genes. The overall data processing and knowledge discovery process is illustrated in Fig. 1 and explained in detail in the next sections.

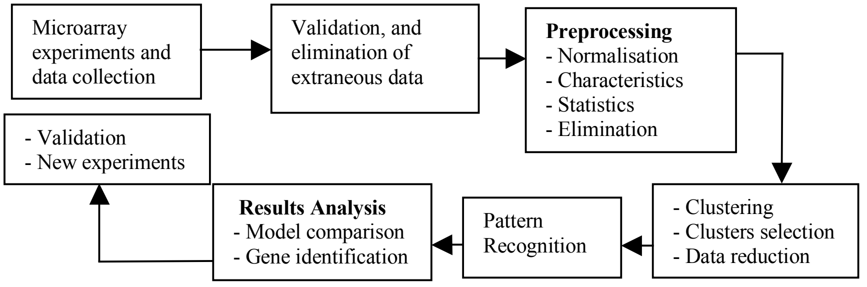


Fig. 1. Knowledge discovery process in microarray data analysis

3 Microarray Experiments, Data Collection, and Bioinformatics Tools

Microarray experiments were performed at the Division of Virology, Children's Hospital of Eastern Ontario. RNA was extracted from the livers of 7 transgenic and 7 non-transgenic mice and analyzed on Mouse cDNA microarrays. Total RNA was isolated using a Qiagen isolation kit (Mississauga, ON, Canada) and used as a template to generate cDNA labeled with Cyanine dye-conjugated (Cy3-green or Cy5-red) dUTP (Amersham Pharmacia). Array images were collected for both Cy3 and Cy5 using a ScanArray XL 4000 fluorescent scanner (Packard Bio-chip, CA) with 10-μm resolution to detect Cy3 and Cy5 fluorescence and image intensity data were extracted and analyzed using QuantArray 3.0 (Packard Bio-chip, CA) software.

The data consisted of seven microarray experiments (biological repeats). Every data set (array) consisted of 30622 rows of readings of mouse genes, and 578 rows of controls. There was one pair (Row 1 and 2, 3 and 4, … 31199 and 31200) of duplicate readings for each gene (clone) or control. Columns specify readings from a non-transgenic mouse (Cy3, Channel 1) and a transgenic (Cy5, Channel 2), respectively. These readings include background, intensity and many other experiment related technical parameters for each channel. Cui and Churchill [3] suggest that for a given number of arrays, more mice per treatment with fewer arrays per mouse is more powerful than fewer mice per treatment with more arrays per mouse. Overall, the amount of data was sufficient to understand the variance across the experiments.

A specialized microarray data pre-processing tool, "Normaliser", was used to perform background subtraction, normalisation and filtering of the raw data from QuantArray. This software is based on the general principles in microarray informatics and is built as an add-in package for Microsoft Excel 2000. We used BioMiner data mining software for the rest of data pre-processing and knowledge discovery experiments reported in this paper. BioMiner has been designed and built in house to provide support for biologists and bioinformaticians performing data mining research in functional genomics. One of the key advantages of the software is that all available forms of data pre-processing and analysis functionalities are integrated into one environment. The data pre-processing and data analysis modules consist of a collection of algorithms and tools to support data mining research activities in an interactive and iterative manner [6 and 16].

4 Data Preprocessing

We started with the raw data from QuantArray that contained experimental results from Channel (Ch1) and Channel 2 (Ch2). For our knowledge discovery studies, only background and intensity columns for each channel of each gene were used. Following a preliminary investigation, Normaliser was used to transform the raw data. The data before and after normalisation were then compared for validation.

4.1 Preliminary Investigation and Validation

This process involved dividing the raw data for each array into Odd and Even subsets representing the duplicates. Using BioMiner, all seven arrays (biological repeats) were examined for their characteristics, e.g. mean and standard deviation. We compared characteristics among the arrays to study the variations and distributions.

Two statistics need to be described, *Skewness* and *Kurtosis*. *Skewness* is a measure of symmetry, or lack of symmetry. A distribution or data set, is symmetric if it looks the same to the left and to the right of the center point (mean). *Kurtosis* shows whether the data is peaked or flat relative to a normal distribution. If the *Kurtosis* is not 0, then the distribution is either flatter (< 0) or more peaked (> 0) than normal.

The examinations of various statistics helped us to understand the characteristics of our data, to identify possible anomalies, and if required to repeat the entire knowledge discovery process after eliminating certain array experiments. In particular, the *Skewness* and *Kurtosis* of one array (ID number: 12230633) were much higher than those of other repeats whose values were close to each other. The particular abnormality of this array will be discussed in detail, in later sections.

4.2 Pre-processing: Background Subtraction, Normalisation, and Filtering

Normalisation, in the context of 2-channel microarray experiments is a transformation to compensate for systematic biases that exist between the two channels of intensity data. The normalisation procedures performed are consistent with the microarray normalisation recommended in [18]. This resulted in 3 new data columns; background subtracted, normalised, Ch1 and Ch2 intensity columns, and a flag column that was used for filtering the data. Filtering removed data points identified as anomalous. All seven microarray experiments were processed by Normaliser using the same settings to generate data sets of background subtracted, log-transformed, normalised and filtered intensity values and corresponding relative gene expression values (Log_2 ratios) for all 31,200 rows (genes and controls). Following are the steps:

1. Background subtraction using the values computed by the quantitation software (QuantArray).
2. Flagging the data for:
   i Spots with intensity in the 5th percentile or lower in either channel.
   ii Spots with intensity in the 98th percentile or higher in either channel.
   iii Spots with intensity/background less than 2.5.
   iv Spots flagged manually during image quantitation.
3. Conversion of intensities to $\log_2$.
4. Pre-filtering the flagged spots for computing the normalisation correction factors.
5. Normalisation correction of Channel-2 intensities using a linear regression of Ch2 vs. Ch1 $\log_2$ intensities by sub-array, such that the slope is 1 and the intercept is 0.
6. Computing the relative gene expression values (Ch2 $\log_2$ intensity – Ch1 $\log_2$ intensity).
7. Filtering of flagged data, if required in (Step 9 below).
8. Averaging (mean) of spot duplicates, if required in (Step 9 below).
9. Assembly of intensity and relative gene expression ($\log_2$ ratio) data sets of all 7 experiments as follows:

i Unfiltered, not averaged
ii Unfiltered, averaged
iii Filtered, averaged

4.3 Additional Processing and Validation

The objective of this step was to analyze the characteristics of the data sets after transformation. The new data sets (Sect. 4.2) were processed accordingly before applying BioMiner software for validation (Fig. 2). Special attention was paid to the filtering (Step 7, Sect. 4.2) since this step influences the research through eliminating some data. Therefore, filtered and unfiltered data sets were separate routes or directions through the knowledge discovery process. The unfiltered data set was split into two directions (or sub-routes). In the first sub-route, the paired spot duplicates (adjacent Odd and Even rows) were averaged as performed on the filtered data. In the second, the spot replicates were split into two separate data sets (Odd and Even). As to this second group (or sub-route), comparative examinations were carried out between the Odd and Even data sets.

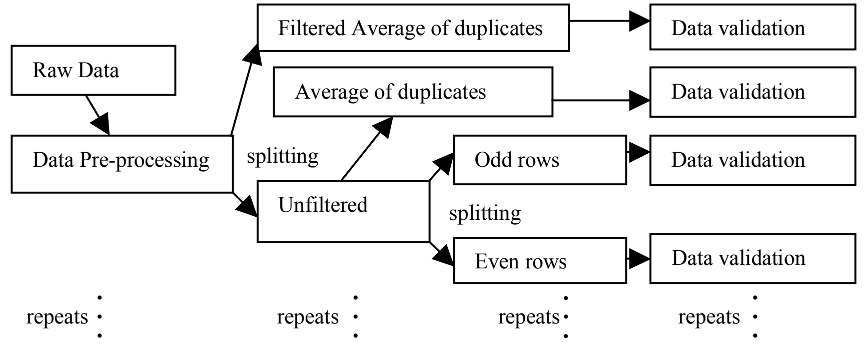


Fig. 2. Data generation strategies after the preprocessing

Using BioMiner software we observed that pre-processing had substantial effects on the data distributions. The histograms of Ch1 and Ch2 showed bell-shaped normal distributions and the scatter plots presented a linear relationship between Ch1 and Ch2. These were consistent with the objectives of pre-processing. The standard deviation of the ratio distribution of one array (ID: 12230633) was two-fold greater than almost all other arrays. This is the same array that showed considerably higher *Skewness* and *Kurtosis* than other repeats (Sect. 4.1). This “abnormal” array was flagged to assess its influence on the knowledge discovery process.

We now had three groups of data (Fig. 2) for knowledge discovery experiments:

i Background subtracted, normalised, filtered data sets of seven arrays with transgenic (Ch2) and non-transgenic (Ch1) readings, and their ratios. For this group, the value for each gene (row) is the average of odd and even rows.
ii As (i) but unfiltered.
iii As (ii) but split into 2 data sets containing odd and even rows, respectively. The contents in this group were similar to the other two groups with two channels.

5 The Knowledge Discovery Process

The knowledge discovery process involved choosing data mining algorithms, selecting suitable options and understanding what to do before taking the next step.

5.1 Unsupervised Learning – Clustering

Based on the objectives of the study listed in section 1, the first strategy was to apply an unsupervised learning method (i.e. using the Clustering module of BioMiner) to identify genes that have certain common properties. This was done on all the data from the 7 arrays. We used this method to (i) group genes based on a similarity or distance measure, (ii) identify and select the most important groups (i.e. up- or down-regulated), and (iii) reduce data dimensionality in order to narrow the search for patterns ("Data Reduction" in Fig. 1). We used K-Means clustering, where K is the number of clusters for each run. Two major routes for clustering are described:

(i) *Clustering genes of each array on values of the two channels*: Filtered, averaged data sets were selected for this analysis. Here, "***difference-in-shape***" was used as the distance measure (Eq. 1). Each cluster is represented as a line connecting Ch1 and Ch2 centroids (average expression value of genes in the cluster). The slope of each line reflects the ratio of Ch2 over Ch1 of that cluster. The significance of regulation was jugged visually based on this slope. For each array, we selected two clusters that contained the most significantly differentially expressed genes (up- or down-regulated: Ch2 vs. Ch1). Then, up- or down-regulated groups of all seven arrays were compared to select the most common genes among them. By experimenting with different values of K (number of clusters) a specific K was chosen such that the final output through this route yielded 41 to 159 most significantly modulated genes.

$$\sqrt{\left|\left(\sum_{j\in A}(X_i[j]-X_k[j])^2\right)-\left(\sum_{j\in A}(X_i[j]-X_k[j])\right)^2 / c\right| / c-1}$$

(Equation 1: difference-in-shape)

$$\left(\sum_{j\in A}(\mathbf{x}_i[j]-\mathbf{x}_k[j])\right)^2 / c^2$$

(Equation 2: difference-in-size)

- $A = \{\, j \mid j \in \{1 .. n\} \wedge$ attribute value $\mathbf{x}_l$[j] is not missing $\}$
- where both X_i [j] and X_k [j] are not missing value, 'c' is the number of variables for which neither X_i [j] nor X_k [j] is missing and 'n' is the total number of variables for certain attribute.

(ii) *Clustering genes on the ratios of all seven arrays together:* Filtered, averaged (sub-route a) and unfiltered, averaged (sub-route b) data sets were studied separately. This time, "***difference-in-size***" was used as the distance measure (Eq. 2). Since ratios from seven arrays were used directly for clustering, the significance of differential expression was also based on the ratios of Ch2 over Ch1. We selected one cluster with the most positive centroid (up-regulated) and the other with the most negative centroid (down-regulated), for two sub-routes (a and b), respectively. The k-value was set in the same manner as in route (i).

5.2 Supervised Learning – Pattern Recognition

To build explicit models from these clusters of genes, classification techniques were used to identify the most informative genes that can discriminate between transgenic and non-transgenic mice. The Pattern Recognition module of BioMiner provides supervised learning techniques including discrimination and prediction algorithms mainly from the WEKA [17] machine learning toolkit. From this collection, the J4.8 Decision Tree induction algorithm [14] was selected to generate tree structures for class assignments. Decision trees are easier to understand and interpret by domain experts, such as biologists, than regression trees (e.g. CART and MART [2, 7 and 9]). Rules can be derived from decision trees. In addition, decision trees are easier to combine with domain knowledge and incorporate into knowledge based systems.

Fig. 3 shows the overall process of searching for patterns. Before applying the decision tree algorithm, three data sets were generated corresponding to the results of clustering described in the previous section (referred to as "Data Reduction" in Fig. 3). In these data sets, rows (14 or 28 cases) were channels (Ch1 and Ch2) of all arrays and columns were genes (between 41 to 159 features). An extra column, containing the label information corresponding to Ch1 or Ch2 (the two classes to be classified), was the last attribute vector in the data. From amongst all attribute vectors (genes), the decision tree algorithm selected genes with the highest information value, which distinguish between the transgenic and non-transgenic mice. The result was a classification model, which included a threshold for classification along with a measure of strength.

5.3 Clustering and Pattern Recognition with Exclusion of One Array

During data processing and validation, one mouse array (biological repeat) was significantly different from the other six (Sect. 4.1 and 4.3). This "abnormality" was also identified in the results of clustering performed using ratios (Sect. 6). Clustering and Pattern Recognition were repeated without this array to see if different genes and models would be generated.

There were no procedural changes for the clustering in Route (i) and Route (ii) (Sect. 5.1), since only the "abnormal" array was excluded. The reduced gene lists (i.e. the most differentially expressed genes, Sect 5.1) identified through clustering may be different, as well as genes (columns) in the corresponding data for pattern recognition (Fig. 3). Also for pattern recognition, the cases of transgenic and non-transgenic mice excluded those from the "abnormal" array in this round. Therefore, there were 12, 12 and 24 rows for Route i, iia and iib respectively (Fig. 3).

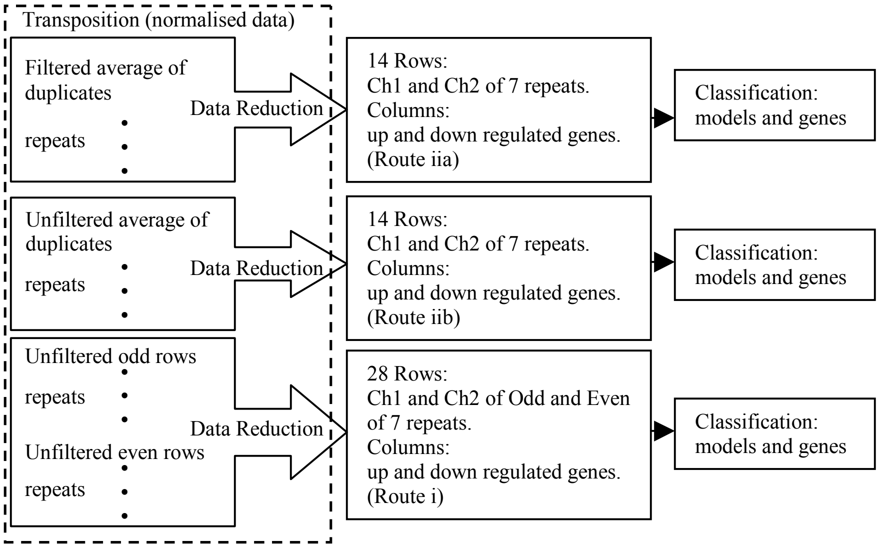

Transposition (normalised data)
Filtered average of duplicates
repeats
Data Reduction
14 Rows:
Ch1 and Ch2 of 7 repeats.
Columns:
up and down regulated genes.
(Route iia)
Classification: models and genes
Unfiltered average of duplicates
repeats
Data Reduction
14 Rows:
Ch1 and Ch2 of 7 repeats.
Columns:
up and down regulated genes.
(Route iib)
Classification: models and genes
Unfiltered odd rows
repeats
Unfiltered even rows
repeats
Data Reduction
28 Rows:
Ch1 and Ch2 of Odd and Even of 7 repeats.
Columns:
up and down regulated genes.
(Route i)
Classification: models and genes

Fig. 3. The data preparation and Pattern Recognition process

6 Results

Using K-Means clustering with a pre-selected number (K) of clusters, we identified the most significant up- and down-regulated genes common to all or most arrays. Table 1 shows the number of identified genes in each of the 6 clustering runs. Runs 1 to 3 include data from all 7 replicate arrays, and runs 4 to 6 exclude data from the "abnormal" array. All clustering runs resulted in a data reduction of between 97.4% and 99.4% and simplified the search for the most informative genes.

Table 1. Results of clustering (* clustering runs that exclude data from one array)

Run #	No. of clusters (K)	Original genes	Genes identified through clustering	Source of data
1	5	12601 (avg.)	71	Filtered channel intensities
2	11	5756	149	Filtered ratios
3	15	15268	159	Unfiltered ratios
4 (*)	5	12601 (avg.)	41	Filtered channel intensities
5 (*)	11	5756	110	Filtered ratios
6 (*)	21	15268	131	Unfiltered ratios

Fig. 4 is an example of clustering performed on ratios of seven arrays (route ii). Each curve is a connection of centroids (Y-axis) of one array across the eleven clusters (X-axis). This shows the "abnormality" of one array (#12230633) that has been emphasized and investigated during processing and validation (Sect. 4). The numbers of genes identified through six runs of clustering, with or without this "abnormal" array were different (Table 1). This demonstrates the influences of including or excluding “abnormal” data on the knowledge discovery process. We have higher confidence in the genes identified when the “abnormal” array was excluded.

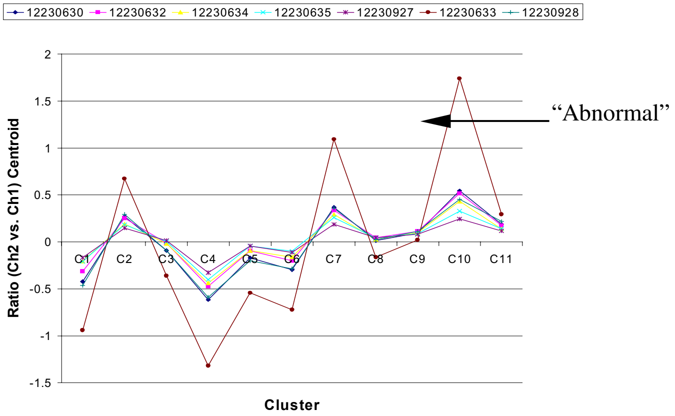


Fig. 4. Clustering patterns and centroids of seven array experiments (route ii)

Following identification of significantly up- or down-regulated genes, we built models that use one or more of these genes to discriminate the two classes (transgenic vs. non-transgenic). The decision tree algorithm generated six fairly simple models from the six sets of significantly regulated genes identified through clustering. All these models achieved 100% classification accuracy and contained between 1-3 genes (Table 2). Fig. 5 shows one of the six models, with two genes. In this model, when gene S57 =< 9.111757, then it is Ch2 (transgenic), otherwise, when S57 > 9.111757 and S101 <= 11.925628 then it is Ch1 (non-transgenic), else, when S57 > 9.111757 and S101 > 11.925628, then it is Ch2.

```
S57 <= 9.111757:          Ch2 (5.0)
S57 > 9.111757
|  S101 <= 11.925628:     Ch1 (7.0)
|  S101 > 11.925628:      Ch2 (2.0)
```

Correctly Classified Instances	14	100%
Incorrectly Classified Instances	0	0%
Total Number of Instances	14	

Fig. 5. An example of classification model

In the last column of Table 2, a total of 8 genes, from amongst all the genes identified through clustering, are used for classification. The first three models (Runs 1 to 3) are quite different based on genes included in their decision trees. With the deletion of the suspected “abnormal” array, the decision trees generated accordingly (Runs 4 to 6) all agree on S2 as one of the most informative genes. Models 5 and 6 are simpler in that they only involve one gene. The consistency (identification of S2 gene) and simplicity of models appear to validate the deletion of the “abnormal” microarray experiment. This gene was also highlighted by some preliminary statistical analyses on these data sets, e.g. t-test with p-value of 0.01.

The Pattern Recognition module may identify additional informative genes via “*discover-and-mask*” approach [6 and 16]. Genes discovered in the decision tree are removed (masked). The remainder of the data is reloaded into BioMiner to generate a second decision tree that reports the next informative genes as nodes. This process may be repeated until (*i*) a drop in the discriminating accuracy of the decision tree, or (*ii*) until none of the remaining genes are able to distinguish the classes.

Table 2. Genes identified in models

Run #	Data Sources – Channels (cases)	Attributes (genes **)	Genes identified
1	Unfiltered odd and even (28) intensities	71 (Run 1)	S43, S17 and S66
2	Filtered average (14) ratios	149 (Run 2)	S57 and S101
3	Unfiltered average (14) ratios	159 (Run 3)	S95 and S66
4 (*)	Unfiltered odd and even (24) intensities	41 (Run 4)	S2 and S63
5 (*)	Filtered average (12) ratios	110 (Run 5)	S2
6 (*)	Unfiltered average (12) ratios	131 (Run 6)	S2
Note: * cases exclude data from one array; ** genes listed in Table1			

7 Conclusions

This paper describes an approach for analyzing large amounts of gene expression data. The objective was to search for meaningful patterns related to discrimination between HCV transgenic and non-transgenic mice. The knowledge discovery experiments performed lead to classification models and the most informative genes. Looking at the classifiers generated, we can see the genes involved, the particular thresholds related to each gene in the model, the relationships (greater than or less than the thresholds), and the strength of these models.

We have tested our method on microarray data of HCV mice experiments. The approach resulted in identification of a small number of the most informative genes, from a total of 15311. From the knowledge discovery point of view, a measure of success is the extent to which the algorithms establish the best models to discriminate different groups. However, from the medical point of view, success is ultimately measured in terms of a prediction and diagnosis of the HCV, especially at the clinical level. The approach proposed in this research has potential for future disease classification, diagnostic and virology applications.

Also emphasized in our approach are the preprocessing, examination, and validation of microarray data before in-depth computation and analysis. These investigations provided us with a clear understanding of the data and resulted in the discovery of an “abnormal” array experiment. Comparison between the results from computations with or without the abnormality further highlighted this discovery. We emphasize that attention should be paid to the results of data quality evaluations, both before and after normalisation. In many studies on microarray data, validation of data quality has not been performed prior to gene discovery analyses.

Acknowledgements. The authors would like to acknowledge the contributions of all members of the BioMine project: Alan Barton, Ziying Liu, Julio Valdes, Youlian Pan and Lynn Wei from IIT, Roy Walker, and Qing Yan Liu from the IBS at NRC and a number of former students who worked in this project. We would also like to recognize the contribution of Dr. Antonio Giulivi and the financial support from Health Canada for this work. Thanks to Bob Orchard for reviewing an earlier version of this paper.

References

1. Bigger, C.B., Brasky, K.M., Lanford, R.E.: DNA microarray analysis of chimpanzee liver during acute resolving hepatitis C virus infection. J. Virology 75 (2001) 7059-7066
2. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., Classification and Regression Trees. Chapman & Hall, New York, (1984)
3. Cui, X., Churchill, G.: How many mice and how many arrays? Replication in mouse cDNA microarray experiments. (In press) Proceedings of CAMDA-02 (2002)
4. Drazan, K.E.: Molecular biology of hepatitis C infection. Liver Transplantation 6 (2000) 396-406
5. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of discrimination methods for the classification of tumors using gene expression data. J. Am. Stat. Assoc. 97 (2002) 77-87
6. Famili, F., Ouyang, J.: Data mining: understanding data and disease modeling. Applied Informatics (2003) 32-37
7. Friedman, J.: Getting started with MART, Tutorial, Stanford University, (2002)
8. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (1999) 531-537
9. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer-Verlag, New York, (2001)
10. Lanford, R.E., Bigger, C., Bassett, S., Klimpel, G.: The chimpanzee model of hepatitis C virus infections. ILAR Journal 42 (2001) 117-126
11. Lanford, R.E., Bigger, C.: Advances in model systems for hepatitis C virus research. Virology 293 (2002) 1-9
12. Li, K., Prow, T., Lemon, S.M., Beard, M.R.: Cellular responses to conditional expression of hepatitis C virus core protein in Huh7 cultured human hepatoma cells. Hepatology 35 (2002) 1237-1246
13. Okabe, H., Satoh, S., Kato, T., Kitahara, O., Yanagawa, R., Yamaoka, Y., Tsunoda, T., Furukawa, Y., Nakamura, Y.: Genome-wide analysis of gene expression in human hepatocellular carcinomas using cDNA microarray: Identification of genes involved in viral carcinogenesis and tumor progression. Cancer Research 61 (2001) 2129-2137
14. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann, San Mateo (1993)
15. Su, A.I., Pezacki, J.P., Wodicka, L., Brideau, A.D., Supekova, L., Thimme, R., Wieland, S., Bukh, J., Purcell, R.H., Schultz, P.G., Chisari, F.V.: Genomic analysis of the host response to hepatitis C virus infection. Proc. Natl. Acad. Sci. USA 99 (2002) 15669-15674
16. Walker, P.R., Smith, B., Liu, Q.Y., Famili, F., Valdes, J.J., Liu, Z.: Data mining of gene expression changes in Alzheimer brain. Data mining in Genomics and Proteomics, a special issue of AI in Medicine, (2004)
17. Witten, I., Eibe, F.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo (1999)
18. Yang, Y.H, Dudoit S., Luu, P., Speed, T.: Normalization for cDNA Microarray Data, SPIE BiOS, San Jose, California, (2001)

Digital Signal Processing in Predicting Secondary Structures of Proteins

Debasis Mitra and Michael Smith

Department of Computer Sciences
Florida Institute of Technology
Melbourne, Florida, USA
{dmitra, msmith}@cs.fit.edu

Abstract. Traditionally protein secondary structure prediction methods work with aggregate knowledge gleaned over a training set of proteins, or with some knowledge acquired from the experts about how to assign secondary structural elements to each amino acid. We are proposing here a methodology that is primarily targeted for any given query protein rather being trained over a pre-determined training set. For some query proteins our prediction accuracies are predictably higher than most other methods, while for other proteins they may not be so, but we would at least know that even before running the algorithms. Our method is based on homology-modeling. When a significantly homologous protein (to the query) with known structure is available in the database our prediction accuracy could be even 90% or above. Our objective is to improve the accuracy of the predictions for the so called "easy" proteins (where sufficiently similar homologues with known structures are available), rather than improving the *bottom-line* of the structure prediction problem, or the average prediction accuracy over many query proteins. We use digital signal processing (DSP) technique that is of *global* nature in assigning structural elements to the respective residues. This is the key to our success. We have tried some variation of the proposed core methodology and the experimental results are presented in this article.

1 Introduction

Since the pioneering works of Anfinsen [1973], it is known that the higher-level structures of proteins are primarily determined by their amino acid sequences. Commonly referred to as the *Protein Folding Problem*, the ability to predict higher-level structures from the sequence remains one of the greatest challenges in bioinformatics [Bourne and Weissig, 2003]. Protein *secondary* structure describes the topology of the chain whereas the *tertiary* structure describes the three-dimensional arrangement of the amino acid residues in the chain. Secondary structures aid in the identification of membrane proteins, location of binding sites and identification of homologous proteins, to list a few of the benefits, and thus highlighting the importance of knowing this level of structure [Rost, 2001]. Experimental methods for structure determination-procedures can be expensive, very time consuming, labor intensive and may not be applicable to all proteins [Brandon and Tooze, 1999] [Rost, 1998]. Spurred by the importance of determining protein structure and the short-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 40–49, 2004.

comings of the laboratory approaches, significant research has been and continues to be devoted to the prediction of the higher-level structures, including the secondary structures, via computational methods.

1.1 Related Works

A simple goal in the secondary structure prediction (termed as Q3) is to determine whether an amino acid residue of a protein is in a *helix*, *strand/sheet* or in neither of the two, in which case the latter is said to be in a *coil* (or *loop*). Published literature for secondary structure prediction spans over a period of four decades. The first generation of secondary structure prediction techniques emerged in the 1960's and were based on single amino acid propensities where, for each amino acid, the probability of its belonging to each of the secondary structural elements is being calculated. The second generation of prediction methods extended this concept by taking into account the local environment, of an amino acid, into consideration. Typically, in predicting the secondary structure for a particular amino acid, information gleaned from segments typically comprising of 3 to 51 adjacent residues were used in the prediction process. Lim's [1974], and Chou & Fasman's [1978] works fall into this category. Prediction accuracies with the second generation methods seemed to saturate at around 60% on an average, seemingly because these methods were local in that only information in a small window of adjacent residues were used in predicting the secondary structure of an amino acid [Bourne and Weissig, 2003]. Local information accounts for only around 65% of the secondary structure information [Rost, 1998]. Since the early 1990's, the third generation prediction methods achieved prediction accuracies around 70%. These methods incorporate *machine learning* techniques, evolutionary knowledge about proteins and use relatively more complex algorithms [Pollastri et al., 2002] [Bourne and Weissig, 2003]. The PHD program [Rost and Sander, 1993] [Rost and Sander, 1994], which uses a system of neural networks, was the first prediction technique to surpass the 70% threshold. Similar performance was later achieved by other systems, which include JPred2 that combines results from various prediction methods. SAM-T99 [Karplus et al., 1998] utilizes Hidden Markov Models and a simple neural network with two hidden layers. SSPro uses bidirectional recurrent neural networks [Bourne and Weissig, 2003] [Baldi and Brunak, 2001].

Most of the above methods target improving the "bottom line" of prediction accuracy, i.e., they target the average accuracy in predicting as many proteins as possible. A great challenge is to improve the accuracy for a query protein for which no significant homologue is available with known structure. However, a user is often interested in reliably high prediction accuracy for his or her particular query protein in hand [see discussions in Rost, 2001] rather than in a prediction server that produces higher prediction accuracies on an average over many proteins. Even when a significant homologue is available these servers tend to produce an average quality result because they are trained on a set of proteins including the ones that are not so-identical to the query. We target this gap in improving the accuracy of prediction for the so called "easy" cases, with available homologues with known structures. With more and more structures available in the PDB the homology-modeling based methods (as ours) are becoming stronger candidates for practical purposes.

1.2. Basic Concepts on DSP-Operators Used

Digital Signal Processing (DSP) is an area of science and engineering undergoing rapid development, largely due to the advances in computing and integrated circuits. In general terms, DSP is the mathematics, algorithms, techniques and methodologies employed in analyzing, manipulating and transforming digital signals or time-series [Openheim and Schafer, 1975]. We map a protein into a digital signal by assigning numeric values to each amino acid.

Our method employs two fundamental DSP operators for predicting secondary structure, namely, the *convolution* and the *deconvolution*. Convolution is a method of combining two signals (typically an *input signal* and a *filter*) to produce a third signal (*output*). Deconvolution is the inverse operation of convolution. Given the output signal and the filter, the input signal may be calculated by deconvolution, or conversely given the input and the output signals, the filter can be calculated.

Mathematically, convolution, between signals x and h, is represented by the following formula, often referred to as the convolution sum,

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Given input signal x and output signal y, the *filter* h can be calculated via deconvolution,

$$H(z) = \frac{Y(z)}{X(z)},$$

where H(z), X(z) and Y(z) are polynomial representations of the impulse response (or the filter), the input signal and the output signal respectively.

We have developed a procedure and conducted a series of experiments to test the viability of the convolution and deconvolution operators in predicting protein secondary structure. The primary and secondary structures of a protein are modeled as input and output signals respectively. The "impulse response" (of the "system" transforming primary structure into secondary structure) or the *filter* is determined by *deconvolution*. The *convolution* of the filter and the primary structure of a protein *t* yields the secondary structure of *t*. The *filter* in essence encodes a black-box type "reasoning" behind how the secondary structure is formed out of a primary structure, as does an artificial neural network (ANN) trained for the same purpose. As in the case of ANN-based systems, we do not intend to further study the filters in order to understand the protein folding problem, rather we just seek for an appropriate filter for the purpose of prediction.

The convolution and deconvolution algorithms are widely available in many DSP toolkits, e.g., MATLAB™, and are routinely used in the signal processing applications. In this sense we are not presenting any new algorithm, rather our work should be considered as a novel procedure for prediction (to be presented later in this article) and its implementation.

1.2.1 DSP-Based Approaches to Other Problems in Bio-informatics

Use of DSP-based methodology is not new in bio-informatics. Its usage is rather gaining some momentum in the field. Elegant representation, efficient algorithms and implementations (even on hardware or firmware) make DSP methods attractive for processing large sequence data

Way back in 1984 Veljkovi et al [1985] conjectured that the biological functions could be guessed by identifying the peak frequencies of some numerical representations of bio-sequences. Although often employed in the analysis of bio-sequences, our literature review indicates an absence of DSP-based approaches for any prediction of structures that we have attempted in our work.

An important question in any DSP-based methods is how to map the biological sequences (e.g., the primary or the secondary structures of proteins for the prediction problem we are addressing) to numerical time-series. In order to transform an amino-acid sequence into a discrete time signal, the numeric *hydrophobicity* value of each amino acid (an amino acid residue's attraction to water molecules) proposed by Kyte and Doolittle [1982], is typically substituted for the corresponding residue in the protein. The justification for using hydrophobicity value is that it is known to be the single most important parameter in affecting the structure of a protein.

We also have an inverse transformation issue that the previously attacked problems using DSP did not have to encounter. The inverse transformation involves mapping the numerical sequences to the corresponding letter-sequences (for secondary structural elements). Our proposed solution is described later.

2 Proposed Formalism

2.1 The Procedure

Our proposed method is a variation of homology-based method of structure prediction. By modeling proteins as discrete-time signals and utilizing the convolution and deconvolution operators, we adopt a rather global view in the secondary structure prediction, i.e., the whole (or at least a long range) of the amino-acid sequence is involved in “deciding” the secondary structural element at any position. The following procedure describes our proposed method.

Suppose the query protein chain is T, whose secondary structure T_s is to be predicted.

Step 1: Perform a PSI-BLAST search, using the primary amino acid sequence T_p of the query protein T. The objective is being to locate a set of proteins, $P = \{P_1, P_2 \ldots\}$ of similar sequences.

Step 2: Select from P the primary structure S_p of a *source* protein, with a significant match to the *query* protein. A PSI-BLAST search produces a measure of similarity between each protein in P and the query protein T, and S_p can be chosen automatically as the protein with the highest such value.

Step 3: Obtain the source protein’s secondary structure, S_s, from the PDB.

Step 4: Align T_p and S_p, to T_{pa} and S_{pa} respectively. Adjust S_s to S_{sa} accordingly.

Step 5: Using S_{pa}, create an input signal S_i (corresponding to the source protein) by replacing each amino acid in the primary structure with its *hydrophobicity* value. The corresponding output signal S_o is created by appropriately replacing the secondary structural elements (SSE’s) in S_{sa} with some numbers.

Step 6: Identify the corresponding filter F_s that transforms S_i to S_o, by performing deconvolution between the two ($F_s = \text{decon}(S_i, S_o)$).

Step 7: Transform the amino acid sequence of the query T_{pa} into a discrete time signal T_i

Step 8: Using the motivation behind the homology modeling techniques that the similar amino acid sequences have similar structures, convolve T_i, with the filter F_s for the source protein, thereby producing the predicted secondary structure (T_o = convolve(T_i, F_s)) of the query protein.

Step 9: The result of this operation T_o is a vector of numerical values. Apply appropriate inverse *translation* (as described in the next subsection) in order to produce the predicted sequence T_{sa} of the corresponding SSE's of the query protein T.

Step 10: Readjust T_{sa} to T_s to be of appropriate length, by *reversing* the effects of alignment originally performed in the Step 4.

2.2 Transformation between Signals and Sequences

As mentioned before the input signal corresponds to an amino acid sequence that replaces each element of the latter by its hydrophobicity value provided by [Kyte and Doolittle 1982]. This scheme seems to be the most reasonable one for the structure prediction purpose.

However, the signal generation corresponding to the sequence of secondary structural elements are not that straight forward. Initially we have experimented with some arbitrary numbers for the three secondary structural elements helix, sheet and coil/other (results are not presented here). Subsequently we have divided the above procedure into two separate ones for predicting helices and sheets independently. The transformations for secondary structural elements (helices or sheets) are Boolean in nature (0 or 1). For example, in the procedure for predicting helices we replace each position (residue) with a helix as 1 and any other type as 0, while converting from the secondary structure to an output signal (e.g., from Ss to So above), and vice versa for reverse transformation (from To to Ts). In an independent run of the experiment for the same query protein, we predict sheets in a similar fashion, and each position in a sheet becomes 1 and the rest are 0's. Subsequently we merge the two predicted sequences (for helices and sheets), replacing any remaining positions as coils in the final predicted structure.

Overlap in the two independently predicted sequences of helices and strands could pose a problem that we did not encounter in the experiments reported here. Heuristics may be deployed in resolving such conflicts. For example, preference would be given to the adjacent elements. Thus, a predicted result of *cchhhhhcc* from the first procedure, and *cccssccc* from the second one could result in ignoring the second prediction process in favor of the first one, because the smaller length of the latter. However, since sheets and helices are rarely adjacent to each other the chance of such erroneous overlapping output from the DSP operators is minimal, as is evidenced in our experiments.

3 Experiments

3.1 Experimental Setup

The filter extracted using the deconvolution (over the source protein) is global in its nature over the respective sequence. The power of using long range influence of the sequence in predicting for a particular location is well known in the literature and are being corroborated with a bidirectional recurrent neural network-based system [Baldi et al, 2001] that drives one of the most accurate secondary structure predicting servers, the SSpro. However, since we apply the filter developed for the source protein, without any modification, to the query protein, the similarity between the two proteins becomes quite important. The major problem here, as with any homology modeling, is to find a suitable source protein in the PDB with a sufficient degree of similarity to the query protein.

For this reason we have also developed a modified procedure by somewhat compromising the global nature of our prediction mechanism. Even when finding a sufficiently similar source protein to the whole query protein is difficult, a much better alignment may be available in the PDB for some segments of the query sequence. In this modified methodology the query protein is segmented into non-overlapping partitions and the above procedure is run independently on each chunk. The resulting predictions for all those chunks are then concatenated. The partitioning is guided from the results of running PSI-BLAST for the whole query sequence. Actually we have experimented with both the non-partitioning methodology (we call that set of experiments as the class N) and the partitioning methodology (class P) for 26 arbitrarily chosen proteins from different families.

3.2 Gap Handling in Alignments

It is known that during evolution mutations may change the individual amino acids in a protein, while the basic structure may remain intact, or at least change at a much slower rate. In order to incorporate this information into the prediction process, instead of inserting and leaving gaps in the aligned sequences, the occurrence of a gap in the query sequence is replaced with the corresponding amino acid in the source sequence. Similarly, gaps in the source sequence are replaced by the corresponding gaps in the query sequence.

For example, say, Tp and Sp are the target and base protein sequence prior to alignment:

Tp : WCSTCLDLACGASRECYDPCFKAFGRAHGKCMNNKCRCYT
Sp : XFTDVKCTGSKQCWPVCKQMFGKPNGKCMNGKCRCYS

The corresponding amino acids in the query sequence, were substituted in the source sequence, as follows:

Tp : WCSTCLDLACGASRECYDPCFKAFGRAHGKCMNNKCRCYT
Bp : XF**S**T**CL**DVKCTGSKQCWPVCKQMFGKPNGKCMNGKCRCYS

4 Results and Discussion

The following Table 1 shows the result over 26 query proteins. The proteins are chosen primarily without any bias. However, we were somewhat careful that they belong to different classes. The secondary structures of the query proteins are actually available, and we measure the accuracy as the ratio (in percentage) of the correctly predicted secondary structural elements to the total number of residues in the chain. The source protein for the query, and the corresponding similarity measure between the two, are being shown in the second and third columns. They are for the non-partitioning (N) methods. For the lack of space we will not show the details of the partitioning schemes, and the corresponding source sequences here. Our results are compared with the prediction accuracies of three of the best/popular servers. The best accuracy from our methods as well as the overall best value for each query is highlighted in each row. The n/a entries indicate we could not perform any experiment.

It is not difficult to analyze why our method outperforms other competing methods for most of the proteins (24 out of 26). Our procedure depends on the best source proteins only, while other methods are typically averaged over a predetermined set of source ones (for the purpose of training or otherwise). Those methods are targeted to provide high average performance over many query proteins, while we concentrate on the proteins that we *know* we could predict well (where we have suitable source protein available in the PDB, say, with >=50% similarity). As the PDB grows fast the chance of finding a good homolg for a query chain increases with time. The primary power of our scheme comes from its global nature (or the long range-nature in the partitioning method).

As expected, a better prediction is made using the global *non-partitioning* (N) method over the other one, namely, the *partitioning* method (P). For low similarity-sources the partitioning method is sometimes better, but not by a large degree.

The average prediction accuracy from Classes N and P were 79% and 80% respectively, with standard deviations of 17% and 10%. The prediction accuracies for SS Pro, PHD and SAM-T99 are 78%, 75% and 77%, with standard deviations of 8%, 8% and 7% respectively. As evidenced by the standard deviations, these methods tend to produce prediction accuracies that are less dispersed or do not exhibit the degree of variability compared to the DSP technique. If we eliminate 5 query proteins for which PDB does not have source proteins with 35% or more similarity, then the average accuracy for classes N and P are 83% and 82%, with the standard deviations 12% and 9% respectively. The same values for the other three methods come to 75%, 76% and 77% average values respectively, with standard deviation values 8% for all those methods. The average and standard deviation values in our two methods are somewhat misrepresentative because only a few query proteins with significantly homologous available source proteins (e.g., query proteins 1PFC and 5TGL over the class N) produced very low scores. Out of 26 query proteins, the class N of the DSP scheme predicted 5 proteins with above 90% accuracy, while none of our competitors crossed that mark for any protein. If we consider only 15 queries with available homologue of 45% and above similarity, then our method's (N) average accuracy is 84% (with the standard deviation 15%), the other three produces respectively, 74%(9%), 76%(9%), and 76%(9%). The same four figures for the six queries with 60% or higher similarity are 93%(3%) for DSP, and 76%(8%), 79%(6%), and

79%(6%) for the other methods. While other methods' accuracy values do not increase much with available better homologues, ours does quite significantly.

Table 1. Prediction accuracy for 26 proteins from PDB

Query Protein	Source Protein (N)	Similarity (N)	Expt Class N Accuracy	Expt Class P Accuracy	PHD Acc.	SSPro Acc.	SAM-T99 Acc.
1A1S	1FV0	46%	**92%**	80%	87%	86%	87%
1CNE	1I7P	48%	**77%**	76%	76%	**78%**	75%
1G7D	1FLM	20%	**19%**	n/a	82%	**84%**	76%
1IH5	1LDF	29%	69%	**71%**	67%	71%	75%
1IJX	1IJY	45%	**94%**	n/a	79%	79%	79%
1IL9	1ABR	40%	**87%**	84%	79%	82%	79%
1LE6	1C1J	35%	82%	**83%**	80%	**87%**	81%
1MKU	1HN4	77%	**93%**	92%	76%	80%	82%
1MVX	1ML9	36%	**79%**	n/a	75%	79%	75%
1PFC	1CQK	47%	63%	**70%**	66%	68%	69%
1PP2	1VAP	71%	**92%**	89%	66%	77%	77%
1QL8	1TRN	75%	**96%**	92%	82%	83%	87%
1VZV	1AT3	48%	**86%**	75%	73%	74%	75%
1XAT	1KK6	38%	**82%**	65%	72%	77%	76%
1KXC	1VCP	64%	**89%**	73%	65%	70%	69%
1J5T	1LBF	35%	83%	**87%**	81%	86%	84%
1TIS	2TSC	45%	**76%**	73%	68%	68%	68%
1B6U	1G0X	33%	74%	**82%**	79%	81%	77%
1GG0	1FXQ	45%	**80%**	75%	80%	86%	85%
5TGL	1LGY	57%	41%	**100%**	51%	51%	50%
1E9I	1PDZ	51%	87%	**90%**	77%	80%	80%
1VDR	1TDR	34%	**80%**	71%	80%	80%	80%
1AI9	1LY3	37%	78%	**80%**	72%	75%	78%
1I74	1K20	80%	**97%**	90%	83%	86%	78%
5RNT	1RDS	64%	**93%**	90%	75%	78%	80%
1IJ9	1GSM	25%	**72%**	61%	73%	**76%**	n/a

5 Conclusion and Future Works

The current philosophy within the automated structure prediction activities is to "achieve as high accuracy for as many queries as possible." This is somewhat elusive because in real life one predicts for one protein at a time [Rost, 2001]. The best *average-accuracy* producing method may not work well for a particular query at hand. In order to address this issue the community is moving toward trying to guess which method will work best for the query at hand and then apply that method. Thus, the structure prediction-servers are sometimes a combination of multiple methods or multiple systems. Our present work is a contribution toward this direction by adding a new method. We have proposed a highly accurate prediction technique for a query where a significantly homologous source protein is available for deriving the predictive filter. We have also proved the worth of utilizing digital signal processing tools in the structure prediction purpose. The bio-informatics community is cautiously exploring the DSP methodologies but have never before used the latter technique for the structure-prediction purpose. In this respect our proposal is a novel one and our results are quite promising.

An obvious direction in our work is to explore why some query proteins with high enough similarity (with the best available source in the PDB) produces low accuracy and vice versa. The primary conjecture here would be a convergent (or divergent, as the case may be) evolutionary distance between the query and the source proteins. This observation may lead us away from using the PSI-BLAST in finding the source protein, and use different similarity metric that are being proposed recently.

Another aspect that we need to study is measuring any difference between the prediction of helices and that of the sheets. We should be doing even better here compared to the other competing techniques in predicting sheets because of the global nature of our scheme. We also need to study how accurate we are in predicting the boundary between the regular structures (helix/sheet) and the non-regular ones (coils). Our technique could be further fine tuned for improved accuracy from this angle of prediction. This is because a signal for only the boundaries (between different structural elements) represents the first derivative of an output signal (corresponding to a secondary structure), which has a higher frequency content than the original one that should be easier to handle by DSP. Instead of predicting three units of structural elements (helix, sheet and coil) we would also like to move to nine units-structure of the DSSP scheme (termed as "output expansion" in [Rost 2001]). Our methodology is also quite suitable for other 1D structure prediction purpose, e.g., the solvent accessibility.

Finally we will enhance the size of our experiments by increasing the number of proteins we have worked with, in order to make our predictive power more reliable. The representative proteins listed in the PDB-list database (http://homepages.fh-giessen.de/~hg12640/pdbselect/) are obvious candidate sets. Trying out different numbering schemes for signal generation from the primary structure (residue to number mapping) is also in our future agenda in this research.

Acknowledgement. This work is partly supported by the National Science Foundation (IIS-0296042).

References

1. Anfinsen, C. B. (1973). "Principles that govern the folding of protein chains." Science. Vol. 181, pp. 223-230.
2. Baldi, P., and Brunak, S. (2001). "Bioinformatics: The machine learning approach." MIT Press.
3. Bourne, P. E., and Weissig, H. (2003). "Structural Bioinformatics." John Wiley & Sons.
4. Brandon, C., and Tooze, J. (1999). "Introduction to Protein Structure." Garland Publishing.
5. Chou, P., and Fasman G. (1978). "Prediction of the secondary structure of proteins from their amino acid sequence." Advanced Enzymology, 47, 45-148.
6. Karplus. K., Barrett, C., and Hughey, R. (1998). "Hidden Markov Models for Detecting Remote Protein Homologies." Bioinformatics, vol. 14, no. 10, 846-856
7. Kyte, J., and Doolittle, R. (1982). "A Simple Method for Displaying the Hydropathic Character of a Protein." Journal of Molecular Biology, 157, 105-132.
8. Lim, V. (1974). "Algorithms for prediction of α-helical and β-structural regions in globular proteins." Journal of Molecular Biology, 88:873-894.
9. Openheim, A. V., and Schafer, R. W. (1975). "Digital Signal Processing." Prentice-Hall, Inc.
10. Pollastri, G., Przybylski, D., Rost, B., and Baldi, P. (2002). "Improving the Prediction of Protein Secondary Structure in Three and Eight Classes Using Recurrent Neural Networks." Protein: Structure, Function and Genetics. 47:228-235
11. Rost B. (1998). "Protein Structure Prediction in 1D, 2D, and 3D." The Encyclopedia of Computational Chemistry (eds. PvR Schleyer, NL Allinger, T Clark, J Gasteiger, PA Kollman, HF Schaefer III and PR Schreiner), 3, 1998, 2242-2255.
12. Rost, B. (2001). "Review: Protein secondary structure prediction continues to rise." Jnl. of Structural Biology, vol. 134, pp. 204-218.

Predicting Protein-Protein Interactions from One Feature Using SVM [*]

Yoojin Chung[1], Gyeong-Min Kim[1], Young-Sup Hwang[2], and Hoon Park[3]

[1] Dept. of Computer Engineering, Hankuk Univ. of Foreign Studies,
89 Wangsan Mohyun, Yongin Kyonggi-do, 449-791, Korea
{chungyj, reboot}@hufs.ac.kr
[2] Division of Computer and Information Science,
[3] Division of Applied Biological Science,
Sunmoon University,
100 Kalsanri, Tangjeongmyeon, Asansi, ChungNam 336-708, Korea
{young, hpark}@sunmoon.ac.kr

Abstract. The interactions between proteins are fundamental to a broad area of biological functions. In this paper, we try to predict protein-protein interactions directly from its amino acid sequence and only one associated physicochemical feature using a Support Vector Machine (SVM). We train a SVM learning system to recognize and predict interactions using a database of known protein interactions. Each amino acid has diverse features such as hydrophobicity, polarity, charge, surface tension, etc. We select only one among these features and combine it to amino acid sequence of interacting proteins. According to the experiments, we get approximately 94% accuracy, 99% precision, and 90% recall in average when using hydrophobicity feature, which is better than the result of previous work using several features simultaneously. Therefore, we can reduce a data size and processing time to 1/n and get a better result than the previous work using n features. When using other features except hydrophobicity, experiment results show approximately 50% accuracy, which is not so good to predict interactions.

1 Introduction

The goal of proteomics is to elucidate the structure, interactions and functions of all proteins within cells and organisms. The expectation is that this gives full understanding of cellular processes and networks at the protein level, ultimately leading to a better understanding of disease mechanisms.

The interaction between proteins is fundamental to the biological functions such as regulation of metabolic pathways, DNA replication, protein synthesis, etc [1]. In biology, it is virtually axiomatic that 'sequence specifies conformation' [5] and it

[*] This work was supported by grant NO. R01-2003-000-10860-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 50–55, 2004.

suggests the following intriguing postulate: knowledge of the amino acid sequence alone might be sufficient to estimate the propensity for two proteins to interact and effect useful biological function. Based on this postulate, Bock and Gough [2] proposed a method to recognize and to predict protein interactions from primary structure and associated physicochemical features using SVM. But Bock and Gough make learning data by concatenating very large amino acid sequence of proteins several times for using several features simultaneously. Therefore, their data size for SVM learning is enormous and the execution time for SVM is very large.

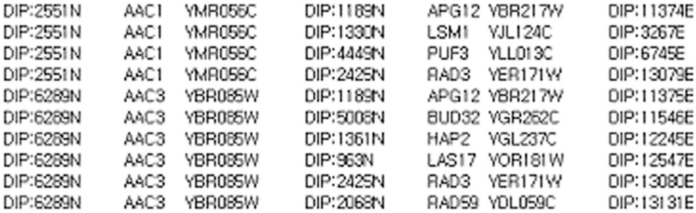
In this paper, we try to predict protein-protein interactions from their amino acid sequences and only one associated physicochemical feature. Each amino acid has diverse features such as hydrophobicity, polarity, charge, surface tension, etc. We select only one among these features and combine it to amino acid sequence of interacting proteins. According to the experiments, we get approximately 94% accuracy, 99% precision, and 90% recall in average when using hydrophobicity feature, which is better than the result of Bock and Gough [2] using several features simultaneously. Therefore, we can reduce a data size and processing time to 1/n and get a better accuracy than the result of Bock and Gough. When using other feature except hydrophobicity, experiment results show approximately 50% accuracy, which is not so good to predict interactions.

2 Methods and Experiments

Our method is as follows.

2.1 Database of Interacting Proteins

Protein interaction data can be obtained from the Database of Interacting Proteins (DIP; http://www.dip.doe-mbi.ucla.edu/). At the time of our experiments, the database comprises 15117 entries representing pairs of proteins known to mutually bind, giving rise to a specific biological function. Here, *interacting* mean that two amino acid chains were experimentally identified to bind to each other. Each interaction pair contains fields linking to other public protein databases, protein name identification and references to experimental literature underlying the interactions. Fig 1 shows a part of DIP, where each row represents a pair of interacting proteins (the first and the fourth columns represent proteins).



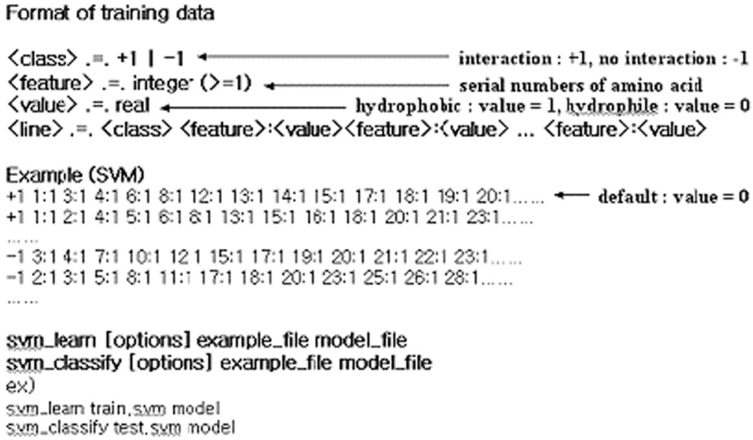
DIP:2551N	AAC1	YMR056C	DIP:1189N	APG12	YBR217W	DIP:11374E
DIP:2551N	AAC1	YMR056C	DIP:1330N	LSM1	YJL124C	DIP:3267E
DIP:2551N	AAC1	YMR056C	DIP:4449N	PUF3	YLL013C	DIP:6745E
DIP:2551N	AAC1	YMR056C	DIP:2425N	RAD3	YER171W	DIP:13079E
DIP:6289N	AAC3	YBR085W	DIP:1189N	APG12	YBR217W	DIP:11375E
DIP:6289N	AAC3	YBR085W	DIP:5006N	BUD32	YGR262C	DIP:11546E
DIP:6289N	AAC3	YBR085W	DIP:1361N	HAP2	YGL237C	DIP:12245E
DIP:6289N	AAC3	YBR085W	DIP:963N	LAS17	YOR181W	DIP:12547E
DIP:6289N	AAC3	YBR085W	DIP:2425N	RAD3	YER171W	DIP:13080E
DIP:6289N	AAC3	YBR085W	DIP:2068N	RAD59	YDL059C	DIP:13131E

Fig. 1. A part of DIP database

2.2 SVM Learning

SVM learning is one of statistical learning theory, it is used many recent bioinformatic research, and it has the following advantages to process biological data. [2]: (1) SVM is computationally efficient [6] and it is characterized by fast training which is essential for high-throughput screening of large protein datasets. (2) SVM is readily adaptable to new data, allowing for continuous model updates in parallel with the continuing growth of biological databases. (3) SVM generates a representation of the non-linear mapping from residue sequence to protein fold space [7] using relatively few adjustable model parameters. (4) SVM provides a principled means to estimate generalization performance via an analytic upper bound on the generalization error. This means that a confidence level may be assigned to the prediction, and alleviates problems with overfitting inherent in neural network function approximation [8].

In this paper, we train an SVM to recognize pairs of interacting proteins culled from the DIP database. The decision rules developed by the system are used to generate a discrete, binary decision (1 = interaction, -1 = no interaction) based on amino acid sequence of the putative protein interaction pair where each value of the sequence is replaced by one of several features of the corresponding amino acid.



```
Format of training data

<class> .=. +1 | −1 ←———————— interaction : +1, no interaction : -1
<feature> .=. intege· (>=1) ←———— serial numbers of amino acid
<value> .=. real ←———————— hydrophobic : value = 1, hydrophile : value = 0
<line> .=. <class> <feature>:<value><feature>:<value> ... <feature>:<value>

Example (SVM)
+1 1:1 3:1 4:1 6:1 8:1 12:1 13:1 14:1 15:1 17:1 18:1 19:1 20:1...... ←— default : value = 0
+1 1:1 2:1 4:1 5:1 6:1 8:1 13:1 15:1 16:1 18:1 20:1 21:1 23:1......
......
−1 3:1 4:1 7:1 10:1 12 1 15:1 17:1 19:1 20:1 21:1 22:1 23:1......
−1 2:1 3:1 5:1 8:1 11:1 17:1 18:1 20:1 23:1 25:1 26:1 28:1......
......

svm_learn [options] example_file model_file
svm_classify [options] example_file model_file
ex)
svm_learn train.svm model
svm_classify test.svm model
```

Fig. 2. An example of the format of training data in TinySVM

In this paper, we use TinySVM (http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM) and an example of its training data format is in Fig 2. We represent an interaction pair by concatenating two amino acid sequences of interacting proteins and by replacing each value of the sequence with a feature value of corresponding amino acid. For example, in the first row of Example(SVM) in Fig 2, 1st,3rd,4th,6th,8th … amino acids of concatenated amino acid sequence of an interacting protein pair have hydrophobicities.

The performance of each SVM was evaluated using accuracy, precision, and recall on the unseen test examples as the performance metrics. The accuracy is defined as the percentage of correct protein interaction predictions on the test set. Each test set consists of nearly equal numbers of positive and negative interaction examples. The precision is defined as the percentage of true positive predictions in all the positive predictions. The recall is defined as the percentage of true positive predictions in all the positive interactions.

2.3 Data Partitioning

For experiments, we divide 12000 entries of protein interactions of yeast in DIP into 12 files as in Table 1. For each experiment, we use 2 files as a training set (one for a positive and the other for a negative interaction sets) and 2 files as a test set (one for a positive and the other for a negative interaction sets). Thus, we use 4000 interactions of yeast in DIP for each experiment.

Table 1. Data partitioning of DIP database for experiments

Lines	File name
1 ~ 1000	1y
1001 ~ 2000	2y
2001 ~ 3000	3y
3001 ~ 4000	4y
4001 ~ 5000	1x
5001 ~ 6000	2x
6001 ~ 7000	3x
7001 ~ 8000	4x
8001 ~ 9000	1w
9001 ~ 10000	2w
10001 ~ 11000	3w
11001 ~ 12000	4w

Testing sets are not exposed to the system during SVM learning. The database is robust in the sense that it represents protein interaction data collected from diverse experiments. There is a negligible probability that the learning system will learn its own input on a narrow, highly self-similar set of data examples [9]. This enhances the generalization potential of the trained SVM.

2.4 Implementation and Experiments

We develop software methods with C++ for parsing the DIP databases, control of randomization and sampling of records and sequences, and replacing amino acid sequences of interacting proteins with its corresponding feature.

To make a positive interaction set, we represent an interaction pair by concatenating two amino acid sequences of interacting proteins and by replacing each value of the sequence with a feature value of corresponding amino acid as in subsection 2.2.

To make a negative interaction set, we replace each value of concatenated amino acid sequence with a random feature value.

We use several features of amino acid such as hydrophobicity, size and polarity. Hydrohobicity feature is motivated by the previous demonstration of sequential hydrophobicity profiles as sensitive descriptors of local interaction sites [10].

Table 2. Experimental results using hydrophobicity.

Model file	Test data set	Accuracy (%)	Precision (%)	Recall (%)
model1y3x	1w3y.svm	97.40	95.40	99.60
	1w4y.svm	97.05	94.76	99.60
	2x3w.svm	99.10	99.59	98.60
	2y4x.svm	99.30	99.20	99.40
model1x4y	1y3w.svm	98.65	100.00	97.30
	1w4y.svm	98.70	100.00	97.40
	2w3x.svm	98.95	99.89	98.00
	2y4w.svm	99.05	99.49	98.60
model2w3y	1w3x.svm	81.80	100.00	63.60
	1y4x.svm	78.91	99.82	57.94
	2x3w.svm	82.75	100.00	65.53
	2y4w.svm	84.00	99.56	68.33
model2x4w	1y3x.svm	99.60	99.79	99.40
	1x4y.svm	98.35	98.88	97.80
	2y3w.svm	98.40	100.00	96.80
	2w4x.svm	99.10	98.61	99.60
Average		94.44	99.06	89.84

Table 2 shows the experimental results using hydrophobicity. In Table 2, model file 'model1y3x' means that 1y file in Table 1 is used as a positive training set and 3x file is used as a negative training set. Test data set '1w3y.svm' means that 1w file is used as a positive test set and 3y file is used as a negative test set. In this experiment, we get approximately 94% accuracy, 99% precision, and 90% recall in average using hydrophobicity feature, which is better than the result of Bock and Gough's using several features simultaneously.

When using polarity feature, experiment results show approximately 50% accuracy, 39 % precision, and 4 % recall in average. When using size feature, experiment results show approximately 55% accuracy, 29 % precision, and 12 % recall in average. Therefore, polarity and size features are not so good to predict interactions.

3 Conclusions

The prediction methodology reported in this paper generates a binary decision about potential protein-protein interactions based on amino acid sequences of interacting proteins and only one associated phycochemical features.

The most difficult thing of our research is to find negative examples of interacting proteins, i.e., to find non-interacting protein pairs. For negative examples of SVM training and testing, we use a randomizing method. But we believe this method makes experimental results worse. Thus finding proper non-interacting protein pairs is important to our future research.

Discovering interacting protein patterns using primary structures of known protein interaction pairs may be subsequently enhanced by using other features such as secondary and tertiary structure in the learning machine.

With experimental validation, further development may produce robust computational screening techniques that narrow the range of putative candidate proteins to those exceeding a prescribed threshold probability of interaction.

Moreover, in the near future, we will parallelize our method on 17-node PC-cluster and get results more fast, which is very important to deal with enormous biological data processing.

References

1. Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J.D.: Molecular Biology of the Cell. 2nd edition, Garland, New York (1989).
2. Bock, J. R. and Gough, D. A.: Predicting protein–protein interactions from primary structure. Bioinformatics, 17 (2001) 455–460.
3. Coward, E. : Shufflet: Shuffling sequences while conserving the k-let counts. Bioinformatics, 15 (1999)1058–1059.
4. Joachims, T. : Making large-scale support vector machine learning practical. In "Advances in Kernel Methods - Support Vector learning", chap. 11, MIT Press, Canbridge, MA, (1999) 169-184.
5. Anfinsen, C.B. : Principles that govern the folding of protein chains, Science, 1 81 (1973) 223-230.
6. Joachims, T. :Making large-scale support vector machine learning practical. In Advances in Kernel Methods-Support Vector Learning, Chap. 11, MIT Press, Cambridge, MA (1999) 169-184.
7. Baldi, P. and Brunak, S.: Bioinformatics: the machine learning approach. In Adaptive Computation and Machine learning. MIT press, Cambridge, MA. (1998).
8. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley, MA (1989).
9. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A. and Nielsen, H. : Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 1, 6 (2000) 412-424.
10. Hopp, T.P. and Woods, K.R.: Predicting of protein antigenic determinants from amino acid sequences, Proc. Natl Acad. Sci. USA, 7 8, 3824-3828 (1981).

Fuzzy OLAP Association Rules Mining Based Novel Approach for Multiagent Cooperative Learning

Mehmet Kaya[1] and Reda Alhajj[2]

[1] Dept of Computer Engineering, Firat University, 23119, Elazığ, Turkey
[2] ADSA Lab & Dept of Computer Science, University of Calgary, Calgary, Alberta, Canada
{mkaya, alhajj}@cpsc.ucalgary.ca

Abstract. In this paper, we propose a novel multiagent learning approach for cooperative learning systems. Our approach incorporates fuzziness and online analytical processing (OLAP) based data mining to effectively process the information reported by the agents. Action of the other agent, even not in the visual environment of the agent under consideration, can simply be estimated by extracting online association rules from the constructed data cube. Then, we present a new action selection model which is also based on association rules mining. Finally, we generalize states which are not experienced sufficiently by mining multiple-levels association rules from the proposed fuzzy data cube. Results obtained for a well-known pursuit domain show the robustness and effectiveness of the proposed fuzzy OLAP mining based learning approach.

1 Motivation and Contributions

Multiagent learning may be modeled by augmenting the state of each agent with the information about other existing agents [8][10]. When the state space of the task is small and discrete, the Q-values are usually stored in a lookup table. But, this method is either impractical in case of large state-action spaces, or impossible with continuous state spaces. One solution is to generalize visited states to unvisited ones as in supervised learning. Functional approximation and generalization methods seem to be more feasible solutions. Unfortunately, optimal convergence of functional approximation for reinforcement learning algorithms has not been proven yet [1][2].

Also, there are several studies cited in the literature where the internal model of each other learning agent is explicitly considered. For instance, Littmann [8] presented 2-player zero-sum stochastic games for multiagent reinforcement learning. In zero-sum games, one agent's gain is always the other agent's loss. Hu and Wellman [5] introduced a different multiagent reinforcement learning method for 2-player general-sum games. However, according to both methods, while estimating the other agent's Q-function, the agent under consideration should observe the other agent's actions and the actual rewards received from the environment. The former agent must know the parameters used in Q-learning of the latter agent. Finally, Nagayuki et al [9] proposed another approach to handle this problem. In their Q-learning method one agent estimates the other agent's policy instead of Q-function.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 56–65, 2004.

Thus, there is no need to observe the other agent's actual rewards received from the environment, and to know the parameters that the other agent uses for Q-learning.

Q-learning has some drawbacks, including modeling other learning agents and experiencing some states more than needed during the learning phase. Although some states are not experienced sufficiently, it is expected that an agent has to select an appropriate action in each state. Also, before the learning process is completed, an agent does not exhibit a certain behavior in some states which may have been experienced sufficiently; hence the learning time is increased. To handle these problems and the like, we propose a novel learning approach that integrates fuzzy OLAP based association rules mining into the learning process. OLAP mining integrates online analytical processing with data mining to substantially enhance its power and flexibility and makes mining an interesting exploratory process.

OLAP is a technology that uses a multidimensional view of aggregated data to provide fast access to strategic information for further analysis. It facilities querying large amounts of data much faster than traditional database techniques, e.g., [3]. Data mining is concerned with the nontrivial extraction of implicit, previously unknown and potentially useful information from data. Discovering association rules is one of the several data mining techniques described in the literature.

Association rules constitute an important class of regularities that exist in databases. A rule $X \rightarrow Y$ is generally rated according to several criteria, none of which should fall below a certain (user-defined) threshold. In common use are the following measures where $D_X = \{T \in D \mid X \subset T\}$ denotes transactions present in database D and contain items in X, and $|D_X|$ is its cardinality. First, a measure of support defines the absolute number or the proportion of transactions present in D and contain $X \cup Y$, i.e., $\sup(X \rightarrow Y) = |D_{X \cup Y}|$ or $\sup(X \rightarrow Y) = \frac{|D_{X \cup Y}|}{|D|}$. Second, the confidence is the proportion of correct applications of the rule, i.e., $conf\ (X \rightarrow Y) = \frac{|D_{X \cup Y}|}{|D_X|}$.

Fuzzy sets provide a smooth transition between members and non-members of a set. Fuzzy association rules are also easily understandable by humans because of the associated linguistic terms. To find out some interesting and potentially useful fuzzy association rules with enough support and high confidence, consider a database of transactions $T = \{t_1, t_2, ..., t_n\}$, its set of attributes I, and the fuzzy sets associated with quantitative attributes in I. Each transaction t_i contains values of some attributes from I and each quantitative attribute in I has at least two corresponding fuzzy sets. We use the following form for fuzzy association rules [7]:

If $X = \{x_1, x_2, ..., x_p\}$ is $A = \{f_1, f_2, ..., f_p\}$ then $Y = \{y_1, y_2, ..., y_q\}$ is $B = \{g_1, g_2, ..., g_q\}$, where A and B contain the fuzzy sets associated with corresponding attributes in X and Y, respectively. Finally, for a rule to be interesting, it should have enough support and high confidence.

The main contributions of the work described in this paper can be summarized as follows: 1) constructing a fuzzy data cube in order to effectively store and process all environment related information reported by agents; 2) estimating the action of the other agent by using internal model association rules mined from the fuzzy data cube; 3) predicting the action of the other agent unseen in the visual environment; 4) selecting the appropriate action of the agent under consideration by using association rules mined from the fuzzy data cube; and 5) generalizing certain states by mining multiple-level association rules from the fuzzy data cube.

The rest of the paper is organized as follows. Section 2 describes a variant of the pursuit problem, to be used as a platform for experiments thought this study. Section 3 includes a brief overview of OLAP technology and introduces the fuzzy data cube architecture. Section 4 presents our learning approach which is based on mining association rules from the constructed data cube. In Section 5, we describe the conducted experiments and discuss the results achieved for the considered environment. Section 6 is the conclusions.

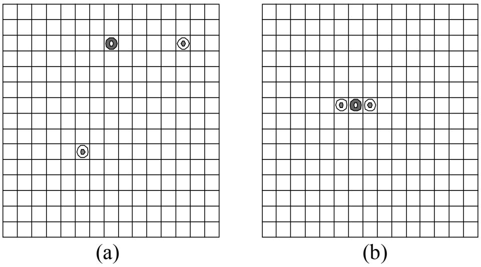


Fig. 1. a) A sample initial position in 15x15 pursuit domain **b)** A sample goal state

2 Pursuit Domain

Samples of the multiagent environment considered in this paper are shown in Fig. 1. It is a variant of the well-known pursuit domain and has the following characteristics. First, it is fully dynamic, partially observable, non-deterministic and has a homogeneous structure. Second, three agents, two hunters and a prey exist in a 15×15 grid world. The initial position of each agent is determined randomly. Third, at each time step, agents synchronously execute one out of five actions: staying at the current position or moving from the current position north, south, west, or east. More than one hunter can share the same cell. However, a hunter cannot share a cell with the prey. Also, an agent is not allowed to move off the environment. The latter two moves are considered illegal and any agent that tries an illegal move is not allowed to make the move and must stay in its current position. Further, hunters are learning agents and the prey selects its own action randomly or based on a particular strategy such as the Manhattan-distance measure. Finally, the prey is captured when the two hunters are positioned at two sides of the prey. Then, the prey and the two hunters are relocated at new random positions in the grid world and the next trial starts.

3 OLAP Technology and Fuzzy Data Cube Construction

A fuzzy data cube is defined based on the fuzzy sets that correspond to quantitative attributes. To understand the whole process, consider a quantitative attribute, say x, it is possible to define at least two corresponding fuzzy sets with a membership function per fuzzy set such that each value of attribute x qualifies to be in one or more of the fuzzy sets specified for attribute x.

Definition 1 (Degree of Membership): Given an attribute x and let $F_x = \{f_x^1, f_x^2, \ldots, f_x^l\}$ be a set of l fuzzy sets associated with x. Membership function of the j-th fuzzy set in F_x, denoted $\mu_{f_x^j}$ is a mapping from the domain of x into the interval [0,1]. Formally, $\mu_{f_x^j} = D_x \rightarrow [0,1]$.

For every value v of x, if $\mu_{f_x^j}(v) = 1$ then v totally belongs to the fuzzy set f_x^j. On the other hand, $\mu_{f_x^j}(v) = 0$ means that v is not a member of the fuzzy set f_x^j. All other values between 0 and 1, exclusive, specify a partial membership degree of v in f_x^j. □

The concept described in Definition 1 is used in building a fuzzy data cube as follows.

Definition 2 (Fuzzy Data Cube): Given an association rules mining task involved with n dimensions, say d_1, d_2, …, d_n of a given data set. The task-relevant data can be pre-computed and materialized into an n-dimensional fuzzy data cube, such that each dimension of the cube contains $\sum_{i=1}^{k} l_i + 1$ values, where k is the number of attributes in dimension X and l_i is the number of membership functions (one per fuzzy set) associated with attribute x_i in dimension X. The last term of the above formula, i.e., "+1" represents a special "Total" value in which each cell stores the aggregation value of the previous rows. These aggregation values show one of the essential features of the fuzzy data cube structure. □

Based on this, we propose a fuzzy data cube to hold the fuzzy information that agents obtain from the environment in real time. Using this cube, we can view the state space of agents from different perspectives. Also, we can perform multiagent fuzzy-OLAP mining on this cube in order to handle the states that have not been experienced sufficiently.

We employed uniform membership functions in representing the state space of agents by fuzzy sets. The definitive intervals of each axis are bounded by [-3, 3], because agent's visual depth is assumed to be 3 in this paper, unless otherwise specified. Here, the location of the prey or the other hunter may be considered in more than one state. For instance, if the hunter observes only the prey at the relative location (-1, -1), then it obtains four different states with equal weight of 0.25. These are [(left, down), not available], [(left, middle), not available], [(middle, down), not available] and [(middle, middle), not available].

The state of a hunter is determined with respect to the x and y directions of the prey and/or the other hunter. For instance, having the prey perceived in the interval [-3, 0] according to the x-direction means that the prey is at the left-hand side of the hunter. Similarly, if the prey is perceived in the interval [0, 3] according to the y-direction, then the prey is above the hunter.

Shown in Fig. 2 is a fuzzy cube with three dimensions, representing the internal model of a hunter; two dimensions of this cube deal with the states of the hunter and the prey in the visual environment, and the third dimension represents the action space of the other hunter. In the fuzzy data cube shown in Fig. 2, the dimensions concerning the state space are values of coordinates x and y, such as [*left*, *down*] and [*left*, *middle*], where *left* shows the x-ordinate, whereas *down* and *middle* represent the y-ordinate. Finally, each cell in the cube shown in Fig. 2 holds what is called *sharing rate*, which is computed based on the observed action of the other agent and states.

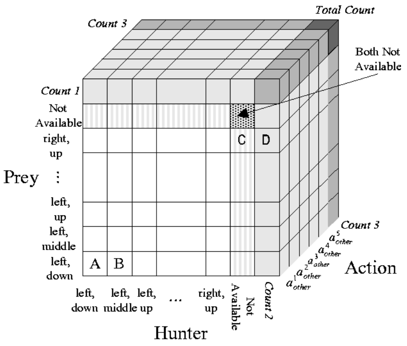


Fig. 2. The proposed fuzzy data cube for the mining process

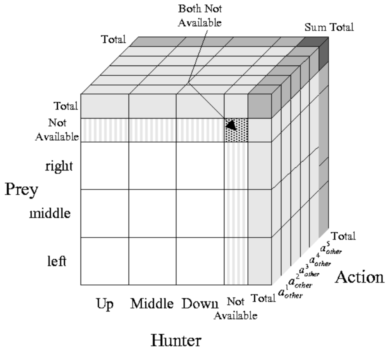


Fig. 3. The cube generated by rolling up some dimensions

Proposition 1 (Sharing Rate Computation): Given a fuzzy data cube with 3 dimensions, if the location of the other hunter has membership degrees $\mu_x(other\ hunter)$ and $\mu_y(other\ hunter)$ along the x and y axes, respectively, the location of the prey has corresponding membership degrees $\mu_x(prey)$ and $\mu_y(prey)$, respectively, and the estimated action of the other agent has membership degree $\mu_{action}(other\ hunter)$, then the sharing rate of all the fuzzy sets to the corresponding cell is computed as: $\mu_{state}(other\ hunter).\mu_{state}(prey).\mu_{action}(other\ hunter)$,where $\mu_{state}(other\ hunter) = \mu_x(other\ hunter).\mu_y(other\ hunter)$ and $\mu_{state}(prey) = \mu_x(prey).\mu_y(prey)$. □

For instance, in case the prey and the other hunter are observed in locations [-2, -2] and [-2, -1], respectively, and the estimated action of the other agent is a^1_{other}, then the values of the cells *A* and *B* in Fig. 2 are both computed as $1\times0.5\times1=0.5$.

4 Employing Association Rules for Multiagent Learning

As mentioned earlier, most of the work already done on multiagent learning assumes a stationary environment, i.e., the behavior of the other agent is not considered in the environment. Whereas it is more natural to consider a dynamic environment in the sense that an agent always learns and each other agent may change its behavior with time too. In such a case, the standard Q-learning approach is not appropriate.

In the work described in this paper, as a given agent executes an action, the other agent's action is also considered. For this purpose, it is necessary to have an internal model database to hold actions of the other hunter. This database, as presented in the previous section, is constructed by employing fuzziness and transformed into a fuzzy internal model data cube structure. This leads to the data cube shown in Fig. 2. In this cube, as long as a hunter observes new states, i.e., as the learning process continues, the observed action of the other agent in the corresponding state(s) is updated. So, as mentioned earlier, each cell of the cube contains the sharing rate calculated with respect to the given state and the observed action. Finally, in order to explicitly express the dependency of the other agent's action, the hunter's Q-function is adjusted as described next in Definition 3.

Definition 3 (Hunter's Q-Function): Given a hunter h_1, which tries to estimate the action of an agent h_2. The corresponding Q-function is represented as $Q(s, a_{self}, a_{other})$, where s is the state that h_1 can observe; $a_{self}(\in A_{self})$ and $a_{other}(\in A_{other})$ are actions of h_1 and h_2, respectively. Here, A_{self} and A_{other} represent sets of all possible actions for h_1 and h_2, respectively. □

According to Definition 3, deciding on whether the action of a given agent is good or not depends on the action of the other agent. In other words, a_{other} is a hidden and major factor in selecting the action a_{self}. In this study, the action a_{other} of the other agent is estimated based on the association rules extracted from the constructed data cube. If one hunter observes the other hunter in its visual environment, then the association rule $s \to a_{other}$ can be easily mined from observations of the other hunter's past actions. On the other hand, if one hunter could not perceive the other hunter, a prediction is done based on the following proposition in order to estimate the direction of the action of the unseen other hunter.

Proposition 2 (predicting the action of the unseen hunter): Given two hunters h_1 and h_2, and a prey P, if h_1 cannot visualize h_2, while P is at a certain location in the visual environment of h_1, then the action of h_2 is predicted based on the general trend of the actions taken by h_2 in the visual environment of h_1 when P was at the same location. □

For instance, assume that the prey is at location [3, 3] and the other hunter is out of the visual environment. In this case, a cell of row C of the data cube shown in Fig. 2 is updated with respect to row D, which shows the number of times the prey visited the location [3, 3], regardless of the position of the other hunter.

To satisfy data mining requirements, the user specifies at the beginning of the learning process a minimum support value for action count, indicated as *count3* in Fig. 2. It is assumed that a state has been experienced sufficiently if its count value reaches this minimum support value. Here, the hunter under consideration, say a_{self}, estimates the action of the other hunter based on the highest confidence value. Otherwise, if a state has not been experienced sufficiently, then a_{self} estimates the action of the other hunter based on the user specified confidence value, say m. If the number of occurrences of a state-action pair is less than m, then this action is not selected in the corresponding state. If there are more actions exceeding the minimum confidence value in a state, then the possibility of selecting an action a_i is computed as: $p(a_i \mid s) = \frac{conf(s \to a_i)}{\sum_{a_j \in A(MinConf)} conf(s \to a_j)}$ where $conf(s \to a_i)$ is the confidence value of the rule $s \to a_i$, and $A(MinConf)$ is the possible set of actions that exceed the minimum confidence value of the corresponding agent.

Next we present our approach of utilizing the constructed fuzzy data cube in mining OLAP association rules that show the state-action relationship. The dimensions of this data cube contain the state information and the actions of both hunters. The mining process utilized for this purpose is described in Algorithm 2.

Algorithm 1: **(Mining Based Multiagent Learning)**

The proposed mining based multiagent learning process involves the following steps:

1. The hunter under consideration observes the current state s and estimates the other hunter's action a_{other} based on the association rules extracted from the fuzzy data cube. If the occurrence number of $s_k \in S$ is greater than the specified minimum support value, then the action a_{other}, having the highest confidence value, is selected.

If the occurrence number of s_k is less than the minimum support value, then the action a_{other} is selected from the actions exceeding the minimum confidence value in state s_k, based on the value of $p(a_i \mid s_k)$.

2. The action a_{self} is selected according to the estimated value a_{other}. In a way similar to the previous association rules mining process, if the count value of a state-a_{other} pair is greater than or equal to the minimum support value determined before, then it is assumed that the relevant state and a_{other} were experienced sufficiently. In this case, the hunter under consideration selects the action with the highest confidence value.
3. If the state-a_{other} pair is not experienced sufficiently, the hunter selects its action with respect to $p(a_i \mid s_k)$; and step 2 is repeated for each component s_k of the observed state *S*. After the action of each component is selected, the action of state *S* is determined as: $\arg \max_{a_{self} \in A_{self}} Q(s_i, a_{self}, a_{other})$
4. The hunter under consideration executes the action a_{self} selected in Step 2 or 3.
5. Simultaneously, the other hunter executes the action a^*_{other}.
6. The environment changes to a new state s^1
7. The hunter under consideration receives a reward *r* from the environment and updates the data cube as follows:

   All $F(s_k, a^*_{other})$ cells are updated in the fuzzy internal model data cube.

   All $Q(s_k, a_{self}, a^*_{other})$ cells are updated using:

   $Q(s_k, a_{self}, a^*_{other}) \leftarrow [1-\alpha.\mu(prey).\mu(other\ hunter)].Q(s_k, a_{self}, a^*_{other}) + \alpha.\mu(prey).\mu(other\ hunter)[r+\gamma. \max_{a'_{self} \in A_{self}} Q(s''_k, a'_{self}, a'_{other})]$ where $a'_{other} = \arg \max_{a'_{other} \in A_{other}} F(s', a'_{other})$, $\mu_{state}(prey) = \mu_x(prey).\mu_y(prey)$, and $\mu_{state}(other\ hunter) = \mu_x(other\ hunter).\mu_y(other\ hunter)$.
8. If the new state s^1 satisfies a terminal condition, then terminate the current trial; otherwise, let $s^l \rightarrow s$ and go back to step 1. □

Experiments showed that after the learning process and based on the escaping policy followed by the prey, some state-action pairs are experienced sufficiently, while others are never visited. The agent selects its own action based on the information obtained from the environment. In some states, this information is not sufficient to mine association rules. In such a case, the information or the data is generalized from low levels to higher levels. To illustrate this, the new data cube shown in Fig. 3 is obtained by rolling up the fuzzy data cube shown in Fig. 2 along the dimensions: *Prey* and *Hunter*. Different strategies for setting minimum support threshold at different levels of abstraction can be used, depending on whether a threshold is to be changed at different levels [4]. We found it more reasonable to use reduced minimum support at lower levels. In fact, most of the work done on OLAP mining considered reduced minimum support across levels, where lower levels of abstraction use smaller minimum support values.

5 Experimental Results

We conducted some experiments to evaluate our approach, i.e., to test the effectiveness of learning by extracting association rules that correlate states and actions. In our experiments, we concentrated on testing changes in the main factors that affect the proposed learning process, namely minimum support, minimum confidence and the multilevel case. Also, to show the superiority of our approach, we compare it with fuzzy Q-learning, denoted FQL in the figures.

All the experiments have been conducted on a Pentium III 1.4GHz CPU with 512 MB of memory and running Windows 2000. Further, in all the experiments, the learning process consists of a series of trials and each reported result is the average value over 10 distinct runs. Each trial begins with a single prey and two hunters placed at random positions inside the domain and ends when either the prey is captured (surrounded by hunters) or at 2000 time steps. Upon capturing the prey, individual hunters immediately receive a reward of 100.

We used the following parameters in the Q-learning process of the proposed approach: learning rate α=0.8, discount factor γ=0.9, the initial value of the Q-function is 0.1, the number of fuzzy sets required to model the states is set to 3, and the visual depth of the agents is set to 3. As FQL is concerned, the following parameters have been used: $\alpha=0.8\rightarrow0.2$, $\gamma=0.9$, the initial value of the Q-function is 0.1; the initial value of the internal model function is 0.2, and the Boltzman distribution has been used for hunters to explore a distinct state.

We performed four different experiments. In the first experiment, the escaping policy of the prey is random, and the minimum confidence value was set at 0% (the best value achieved in Fig. 5) until the number of occurrences of each state reached the minimum support value. Fig. 4 shows the learning curves of the steps required to capture the prey for different values of minimum support. As can be easily seen from Fig. 4, the learning curve for the case when minimum support value was set to 4K (denoted MinSup4K in Fig. 4) converges to the near optimal solution faster than both MinSup6K and FQL.

Here, it is worth mentioning that the three values plotted in Fig. 4 have been selected based on extensive analysis of different minimum support values. From the analysis, we realized that as we increased the minimum support value beyond 4K, the curves started to follow a trend similar to the curve obtained when the minimum support value was 4K. This is obvious from comparing the two curves MinSup4K and MinSup6K. Finally, similar analysis was conducted in selecting the other values plotted in the rest of the figures presented in this section.

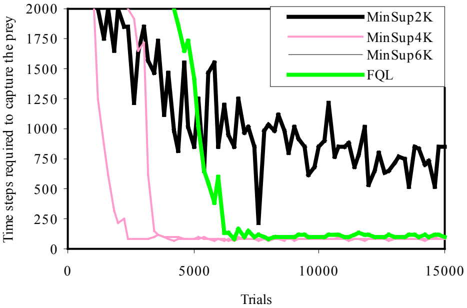


Fig. 4. Learning curves of hunters when the prey escapes randomly; with minimum confidence fixed at 0%

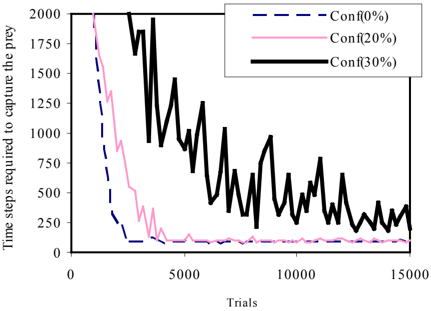


Fig. 5. Learning curves of the hunters with minimum support fixed at 4K

In the second experiment, we fixed the minimum support and checked the learning process for different values of minimum confidence. Plotted in Fig. 5 are the learning curves for different values of minimum confidence with the minimum support fixed at 4K. Discovering the environment is important up to a particular level of confidence value, i.e., the agent should be given the opportunity to discover its environment. Further, the solution when the confidence value is set to 0% (labeled Conf(0%) in Fig. 5) not only converges faster, but also captures the prey in less number of steps.

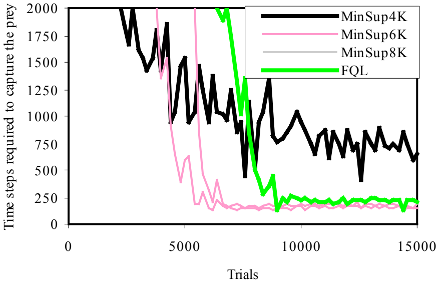


Fig. 6. Learning curves of the hunters when the prey escapes using Manhattan-distance

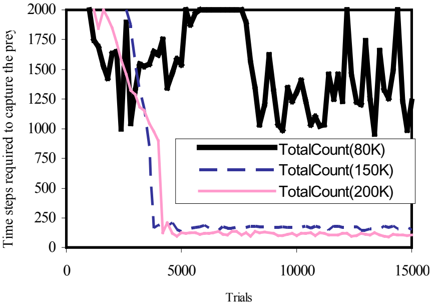


Fig. 7. Learning curves of the hunter with generalized states

In the third experiment, the prey escapes based on the Manhattan-distance to the located hunter. Comparing the results of this experiment as shown in Fig. 6 with the results of the previous two experiments, it can be easily observed that each state has to be experienced more for more complex multiagent environments. Further, it is enough to select the minimum support value as 6K because the hunter cannot learn for lower minimum support values, and it is unnecessary to increase the minimum support value beyond 6K, which also outperforms FQL.

The last experiment is dedicated to the multiple levels case with the prey escaping using Manhattan-distance. Our goal is to generalize a state that has not been experienced enough compared to other states. In such a case, the hunter observes the environment from a higher level and decides on the best action. Here, if the total count reaches the pre-determined threshold value and the minimum support value of the current state is below the threshold, then the hunter goes up to a higher level in order to get more detailed information from the environment. The results of this experiment for three different total count values are plotted in Fig. 7.

6 Conclusions

In this paper, we proposed a novel multiagent reinforcement learning approach based on fuzzy OLAP association rules mining. For this purpose, we start by embedding the fuzzy set concept into the state space in order to decrease the number of states that an agent could encounter. Then, we defined a fuzzy data cube architecture for holding all environment related information obtained by agents. Using this cube effectively, we extracted fuzzy association rules from the past actions of agents. Based on these rules, we handled two important problems that are frequently faced in multiagent learning.

First, we estimated the action of the other agent, even when it is not in the visual environment of the agent under consideration. Second, we presented a new action selection method in order for the agents to take the most appropriate action. For this purpose, we generalized the states that were not experienced sufficiently. In fact, multiagent learning is a very difficult problem in general, and the results obtained may depend on specific attributes of the problem. However, experimental results obtained on a well-known pursuit domain showed that the proposed fuzzy OLAP mining based learning approach is promising for emerging adaptive behaviors of multiagent systems. Currently, we are investigating the possibility of applying our method to more complex problems that require continuous state space and to develop and improve different corresponding algorithms.

References

1. Abul O., Polat F. and Alhajj R., "Multiagent reinforcement learning using function approximation," *IEEE TSMC*, Vol.30, No.4, pp. 485-497, 2000.
2. Baird L., "Residual algorithms: Reinforcement learning with function approximation," *Proc. of ICML*, pp.30-37, 1995.
3. Chaudhuri S. & Dayal U., "An overview of data warehousing and OLAP technology," *ACM SIGMOD Record*, Vol.26, pp.65-74, 1997.
4. Han J. and Fu Y., "Mining multiple-level association rules in large databases," *IEEE TKDE,* Vol.11, No.5, pp.798-804, 1999.
5. Hu J. and Wellman M.P., "Multiagent reinforcement learning: theoretical framework and an algorithm," *Proc. of ICML,* pp.242-250, 1998.
6. Kaya M. and Alhajj R., "Reinforcement Learning in Multiagent Systems: A Modular Fuzzy Approach with Internal Model Capabilities," *Proc. of IEEE ICTAI*, pp.469-475, 2002.
7. Kuok C.M., Fu A.W. and Wong M.H., "Mining fuzzy association rules in databases," *SIGMOD Record*, Vol.17, No.1, pp.41-46, 1998.
8. Littman M.L., "Markov games as a framework for multi agent reinforcement learning," *Proc. of ICML*, pp.157-163, 1994.
9. Nagayuki Y., Ishii S. and Doya K., "Multi-Agent reinforcement learning: An approach based on the other agent's internal model," *Proc. of IEEE ICMAS*, pp.215-221, 2000.
10. Tan M., "Multi-agent reinforcement learning: independent vs cooperative agents," *Proc. of ICML,* pp.330-337, 1993.

OIDM: Online Interactive Data Mining *

Qijun Chen, Xindong Wu, and Xingquan Zhu

Department of Computer Science, University of Vermont, Burlington VT 05405, USA
{qchen, xwu, xqzhu}@cs.uvm.edu

Abstract. Facilitated by the achievements of various data mining techniques, both academic research and industrial applications are using data mining tools to explore knowledge from various databases. However, building a mining system is a nontrivial task, especially for a data mining novice. In this paper, we present an online interactive data mining toolbox – OIDM, which provides three categories (classification, association analysis, and clustering) of data mining tools, and interacts with the user to facilitate his/her mining process. The interactive mining is accomplished through interviewing the user about his/her data mining task. OIDM can help the user find the appropriate mining algorithm, refine the mining process, and finally get the best mining results. To evaluate the system, the website of OIDM (2003) has been released. The feedback is positive. Both students and senior researchers found that OIDM would be useful in conducting data mining research.

1 Introduction

Advances in database technologies and data collection techniques including barcode reading, remote sensing, and satellite telemetry, have incurred the collection of huge amounts of data. Generally, the information in a database can be divided into two categories: explicit information and implicit information. Explicit information is the information represented by the data while implicit information is the information contained (or hidden) in the data. For example, in a relational database, a tuple in a "student" table represents explicitly the basic information about a student. Other information items, such as the relationships between tables and the dependencies between attributes are also documented in the database. All of them are considered as explicit information. Traditional database retrieval techniques are used to get this information. There is also implicit information. For example, the associations between the birth months of the students and their exam scores are information items that are implicit but useful. Those information items can be discovered (or mined) but cannot be retrieved. Data mining techniques are developed for this purpose. Such an operation is referred to as *data mining* or *knowledge discovery in databases* (KDD) (Agrawal and Srikant, 1994; Cendrowska, 1987; Fisher, 1987; Holte, 1993; Quinlan, 1993; Wu, 1995). Data mining can be defined as the *discovery of interesting, implicit,*

* This research is supported by a NASA EPSCoR grant.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 66–76, 2004.

and previously unknown knowledge from large databases (Fayyad *et al.*, 1996). It involves techniques from machine learning, database systems, data visualization, statistics, and information theory.

Two questions need to be answered in order to perform a data mining task effectively and efficiently. 1) For a specific data set, what is the most suitable data mining algorithm? Nowadays, various algorithms have been developed to deal with different problems (such as classification, clustering, and association mining). Even classification could imply very many different algorithms, such as C4.5 (Quinlan, 1994), CN2 (Clark P., and Niblett, 1989), and HCV (Wu, 1995). This question becomes even more difficult for a data mining novice. 2) How could the user be actively and interactively involved in the mining process? Since the background knowledge from the user is crucial to the usefulness of the final mining results. Unfortunately, even though research in data mining has made substantial progresses, rare efforts have been made to solve these critical issues.

In this paper, we design an online data mining toolbox – OIDM, which provides three categories (classification, association analysis, and clustering) of data mining tools and interacts with the user while performing data mining tasks. OIDM combines normal functions of an expert system: asking questions, integrating evidence, algorithm recommendation, and summarizing the results. The interactive property is accomplished through interviewing the user and integrating the feedback from the user. OIDM can iteratively and progressively help the user find the best mining results for his/her data mining tasks. Initially the user may have no knowledge about either data mining or what can be discovered from his/her data. By interacting with the user and analyzing his/her answers to a set of well-designed questions, OIDM can gradually refine the user requirements and fulfill the task. OIDM is particularly useful for data mining beginners and can also facilitate data mining experts in their data mining research.

Instead of developing a new mining algorithm, OIDM is constructed on existing data mining algorithms, as our goal is to free the user from programming and to involve the user into an active mining process. OIDM has the following features:

- It's a programming-free toolbox. No programming work is required from the user.
- The interactive mechanism involves the user into a deeper level of the system during the mining process.
- Multi-layer result summarization presents the mining results in a progressive way, which helps the user in interpreting the mining results.

2 Related Work

Interactive data mining is not a new concept, especially when data mining is perceived from the statistical point of view (Hand, 1994). To select an optimal learning algorithm for a certain task, two popular mechanisms exist: 1) One approach is to learn a decision tree for the applicability of the available algorithms based on the data characteristics (Brazdil et al., 1994), and 2) Another approach is a user-centered mechanism used in the Consultant part of the MLT-project (Craw, 1992). The survey (Verdenius 1997), which dealt with the question of how companies can apply inductive learning techniques, concluded that the process of machine learning should primarily be user-driven, instead of data- or technology driven. Such a conclusion can also be found among many other papers (Brodley and Smyth 1995).

In the early 1990s, several researchers at the University of Aberdeen conducted a research project – CONSULTANT (Craw, 1992; Graner et al., 1992; Kodratoff et al., 1992). CONSULTANT is employed to help the user find the best classification tool for a specific dataset. CONSULTANT questions the user about the task to be solved, gathers data and background knowledge, and recommends one or more learning tools. However, with an interactive mechanism, CONSULTANT can only deal with the classification problems.

To facilitate knowledge acquisition, a model needs to be predefined in the toolbox. This is generally acknowledged in the knowledge elicitation community: "The main theories of knowledge acquisition are model-based to a certain extent. The model-based approach covers the idea that abstract models of the tasks that expert systems have to perform can highly facilitate knowledge acquisition" (Heijst et al, 1992). To enhance the flexibility of a CONSULTANT-like mechanism in model construction, White and Sleeman (1998) introduced MUSKRAT (Multistrategy Knowledge Refinement and Acquisition Toolbox), which includes an advisory system coupled with several knowledge acquisition tools and problem solvers. MUSKRAT compares the requirements of the selected problem solver with the available sources of information (knowledge, data, and human experts). As a result, it may recommend either reusing the existing knowledge base, or applying one or more knowledge acquisition tools, based on their knowledge-level descriptions.

Although helpful in involving the user into the mining process, the above techniques only address data mining problems through machine learning techniques. To broaden the meaning of interactive mining, other research efforts have been made, in which interactive mining can be facilitated by visualization techniques (Ware, 2001), active data mining (Motoda, 2002), or decomposing a problem into subtasks where different mining mechanisms could be involved (Robert, et al., 1997).

Ware (2001) proposed a graphical interactive approach to machine learning that makes the learning process explicit by visualizing the data and letting the user "draw" decision boundaries in a simple but flexible manner. A similar research effort can be found in Hellerstein (1999). However, even though these visualization techniques could make data mining more intuitive, it may decrease the mining efficiency in handling realistic problems where data mining could be very complicated and involve different mining mechanisms. To address this problem, Robert et al. (1997) proposed an approach which involves systematically decomposing a data mining process into subtasks and selecting appropriate problem-solving methods and algorithms. A similar problem in statistics has been conducted by Hand (1994).

In the OIDM project, we adopt a CONSULTANT-like mechanism to facilitate interactive data mining, and an interaction model is defined in advance. The reasons for using a predefined model are as follows:

1. It's simple for system management. Adding a new data mining algorithm can be accomplished by minor modifications in the system model.
2. It can help generate a compact solution for interactive mining. Though interactive, an efficient system should not require the user to answer dozens of questions before he/she can get the results. Being cooperative is the users' willingness but not his/her responsibility. To be practical, the interactive process should be as compact as possible, which means it can guide the user to achieve what they want in a few steps.

Although OIDM is similar to CONSULTANT, there are two differences between them. First, OIDM provides a broader range of data mining tools, which cover classification, clustering and association analysis. Second, the goal of OIDM is to help the user find the best learning result not just the best learning tool.

3 System Design

To design an interactive data mining toolbox, which is practical and efficient in handling realistic problems, three goals need to be achieved

- **Interactive.** Interaction between the user and the system is the channel through which the system can collect information from the user. Furthermore, it is a good way to let the user know more about the mining task and the underlying data mining algorithms.
- **Complete.** To be a complete solution, the system must be able to collect all the necessary information from the user before the algorithm selection. The user should be provided with as many choices as possible for each question.
- **Compact.** To be a compact solution, the system should only post the indispensable questions to the user. A compact design should make the data mining process as intuitive as possible.

3.1 System Workflow

The system framework of OIDM is shown in Figure 1. It runs by following a predefined model. First, OIDM recommends to the user one specific mining algorithm through the Algorithm Selection Module (Section 3.2). Once the algorithm is selected, OIDM asks the user to provide input data. The user can choose to upload data files or paste the data in the given text areas on OIDM. Based on the input data, OIDM constructs the input files, which conform to the selected algorithm through the Data Processing Module (Section 3.3). After the data processing stage, OIDM runs the mining algorithm on the input data and provides the user with the results through a Multi-level Summarization Mechanism (Section 3.4). The user may find the results not satisfactory. To refine the results, the user can choose to tune the parameters through the Parameter Tuning Module (Section 3.5) or select a different algorithm.

By doing this, OIDM can not only guide the user to select the right mining tool, but also provide experimental comparisons between different mining mechanisms or different parameter settings of the same algorithm. OIDM will follow this iterative workflow until useful mining results are found.

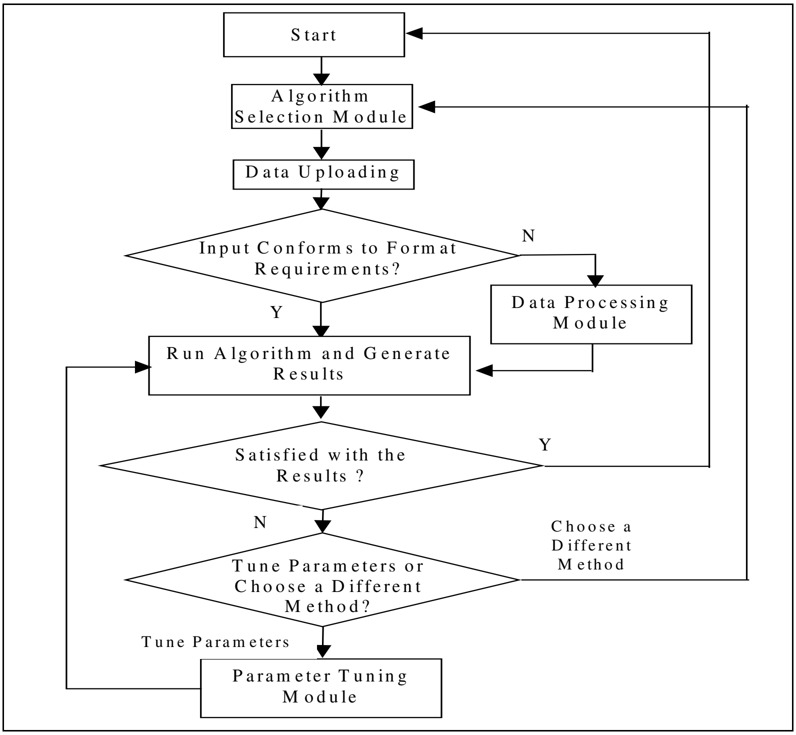

Start
Algorithm Selection Module
Data Uploading
Input Conforms to Format Requirements?
N
Y
Data Processing Module
Run Algorithm and Generate Results
Satisfied with the Results ?
Y
N
Choose a Different Method
Tune Parameters or Choose a Different Method?
Tune Parameters
Parameter Tuning Module

Fig. 1. System workflow of OIDM

3.2 Algorithm Selection Module

OIDM consists of the following seven typical mining algorithms, which cover three popular categories of data mining problems: Classification, Clustering and Association Analysis. In this section, we introduce the interaction model that is used to guide the user in selecting a mining algorithm. The functionalities of the system can be easily extended through adding more mining algorithms.

- **C4.5** (Quinlan, 1993)**:** A decision tree construction program.
- **C4.5Rules** (Quinlan, 1993)**:** A program that generates production rules from unpruned decision trees.
- **HCV** (Wu, 1995)**:** An extension matrix based rule induction algorithm.
- **OneR** (Holte, 1993): A program that constructs one-level rules that test one particular attribute only.

- **Prism** (Cendrowska, 1987)**:** An algorithm for inducing modular rules.
- **CobWeb** (Fisher, 1987)**:** An incremental clustering algorithm, based on probabilistic categorization trees.
- **Apriori** (Agrawal and Srikant, 1994)**:** An algorithm for mining frequent itemsets for boolean association rules.

These seven algorithms are organized into a hierarchy, as shown in Figure 2, to help the user clarify their mining task. Algorithm selection follows this hierarchy. If the user has no knowledge about data mining or is not sure about which algorithm is the most suitable, the system can help him/her choose one through providing some typical mining tasks and asking the user to choose a similar one.

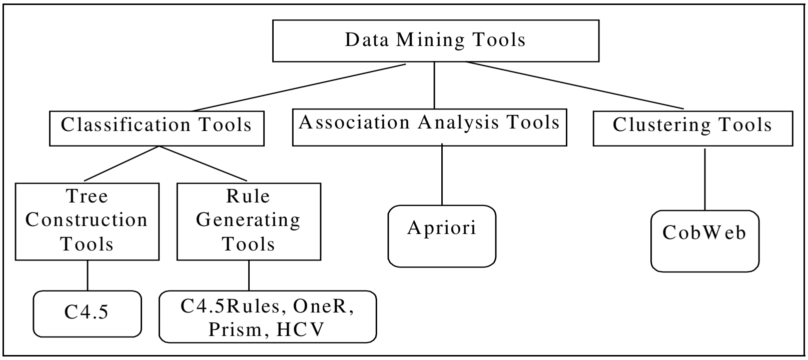


Fig. 2. Algorithm selection model

3.3 Data Processing Module

All data mining software packages require the input data follow a specific data format (such as csv – a comma delimited format) before the algorithm can actually run on the given datasets. Furthermore, most algorithms require the user to provide some domain knowledge for the raw data such as what are the possible values for a particular nominal attribute. OIDM provides the Data Processing Model (DPM) to help the user. DPM can extract domain knowledge automatically from the input data and ask the user to refine the domain knowledge if necessary. Through asking the user a serial of common questions (such as which attribute will be treated as the class label, and whether a specific attribute is nominal or continuous), DPM can convert the original data file (if the field delimiter is other than the comma) and construct input files that meet a specific algorithm's input format automatically. The only input from the user would be the data files and answers for some specific questions. Two common input files can be generated through DPM: arff (for the WEKA package) and names&data (for C4.5 and HCV).

3.4 Multi-level Summarization Module

Summarizing the mining results can be very useful, because some mining mechanisms generate complicated results. For example, it's quite common to generate more than 100 classification rules in a typical classification problem. Consuming the whole results at one time is unnecessary sometimes. Consequently, instead of showing the whole details of the mining results, we use a Multi-level Summarization Model, which provides the user with two levels of results. At the first level, general information of the mining results is provided, such as the classification accuracy, the coverage of the results, and the statistical information. From this information the user can easily get the performance information of the selected algorithm. If the user is particularly interested in a specific result, he/she can drill into the second-level results which include the details of the mining results, such as the classification rules, clusters, and the coverage and accuracy of the association rules. By using the multi-level summarization module, the system can be more informative and practical.

3.5 Parameter Tuning Module

To improve the mining results, OIDM provides two types of interactions: 1) selecting different types of mining algorithms, and 2) setting different parameters for a specific algorithm (Figure 1). Experienced users usually know which mining algorithm to use before launching OIDM. Therefore, parameter tuning is more useful to them. OIDM provides a unique set of common parameters for each mining algorithm. Consequently, different parameter tuning options are presented based on the selected algorithm. Brief explanations for the parameters are also provided. The default parameter values are provided initially. For detailed explanations of the parameters, the user may refer to the online algorithm manuals, which are also linked on OIDM.

4 An Example Run of OIDM

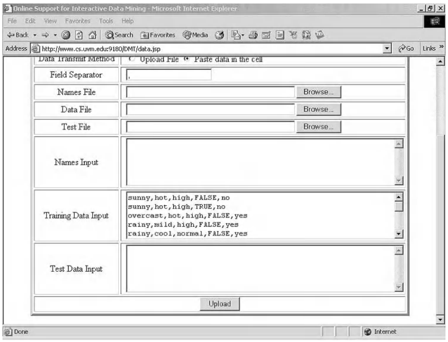
In this section, we present a demonstration where OIDM is used to solve a classification problem.

1. Start Trip:
2. **Q**: Which data mining tool would you like to use?
   (a) Clustering; (b) Classification; (c) Association Analysis; (d) Not Sure
   A: Not Sure
3. **Q**: Which of the following categories of problems is your problem similar to?
   (a) Segment a customer database based on similar buying patterns.
   (b) Find out common symptoms of a disease.
   (c) Find out whether customers buying beer will always buy diapers.
   (d) None of the above.
   A: (b) Find out common symptoms of a disease.

4. **Q**: Would you prefer the output in the form of:
   (a) A Decision Tree; (b) A Set of IF-THEN Rules
   A: (a) A Decision Tree

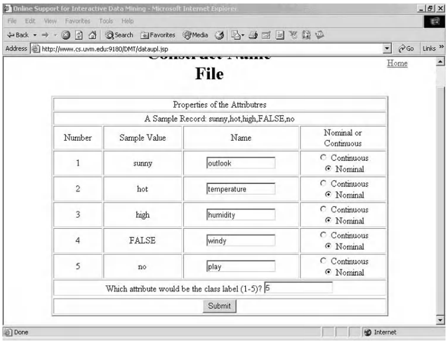
5. **Q**: Which classification tool would you like to use?
   (a) C4.5
   A: C4.5

6. Data Uploading Page (Figure 3)


Online Support for Interactive Data Mining - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cs.uvm.edu:9180/DMT/data.jsp
Go Links
Done
Internet

Data Transmit Method	○ Upload File ● Paste data in the cell
Field Separator	,
Names File	Browse...
Data File	Browse...
Test File	Browse...
Names Input	
Training Data Input	sunny,hot,high,FALSE,no sunny,hot,high,TRUE,no overcast,hot,high,FALSE,yes rainy,mild,high,FALSE,yes rainy,cool,normal,FALSE,yes
Test Data Input	
Upload	

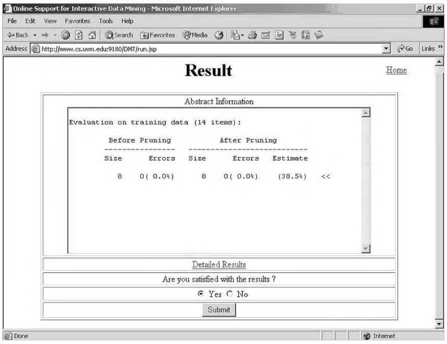
Fig. 3. Data uploading page


Online Support for Interactive Data Mining - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cs.uvm.edu:9180/DMT/dataupl.jsp
Go Links
Home
Done
Internet

Construct Name File

Properties of the Attributres			
A Sample Record: sunny,hot,high,FALSE,no			
Number	Sample Value	Name	Nominal or Continuous
1	sunny	outlook	○ Continuous ● Nominal
2	hot	temperature	○ Continuous ● Nominal
3	high	humidity	○ Continuous ● Nominal
4	FALSE	windy	○ Continuous ● Nominal
5	no	play	○ Continuous ● Nominal
Which attribute would be the class label (1-5)? 5			
Submit			

Fig. 4. Data processing page


Online Support for Interactive Data Mining - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cs.uvm.edu:9180/DMT/run.jsp
Go Links
Home
Done
Internet

Result

Abstract Information

```
Evaluation on training data (14 items):

         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate

           8    0( 0.0%)      8    0( 0.0%)    (38.5%)   <<
```

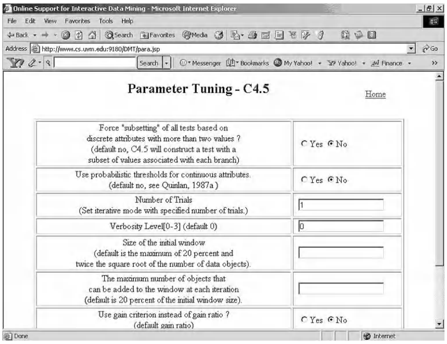
Detailed Results

Are you satisfied with the results ?

● Yes ○ No

Submit

Fig. 5. Multi-level result page


Online Support for Interactive Data Mining - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cs.uvm.edu:9180/DMT/para.jsp
Go
Search Messenger Bookmarks My Yahoo! Yahoo! Finance
Home
Done
Internet

Parameter Tuning - C4.5

Force "subsetting" of all tests based on discrete attributes with more than two values ? (default no, C4.5 will construct a test with a subset of values associated with each branch)	○ Yes ● No
Use probabilistic thresholds for continuous attributes. (default no, see Quinlan, 1987a)	○ Yes ● No
Number of Trials (Set iterative mode with specified number of trials.)	1
Verbosity Level[0-3] (default 0)	0
Size of the initial window (default is the maximum of 20 percent and twice the square root of the number of data objects).	
The maximum number of objects that can be added to the window at each iteration (default is 20 percent of the initial window size).	
Use gain criterion instead of gain ratio ? (default gain ratio)	○ Yes ● No

Fig. 6. Parameter tuning page

7. Data Processing Page (Figure 4)

8. Result Page (Figure 5)

9. **Q**: Are you satisfied with the results?
   (a) Yes; (b) No
   A: (b) No

10. **Q**: Choose a different method?
    (a) Choose a different method; (b) Tune Parameters
    A: (b) Tune Parameters

11. Parameter Tuning Page (Figure 6)

12. Result Page (Figure 5)
    Q: Are you satisfied with the results?
    (a) Yes; (b) No
    A: (a) Yes

13. Start Page

5 System Evaluation

To evaluate the system, we have released the website of OIDM online and collected the feedback from the users. Basically, all suggestions conclude that OIDM is a useful toolbox. It is expected that two types of people would be particularly interested in OIDM: 1) Students who have taken a data mining or artificial intelligence course and are interested in conducting research in data mining. Their feedback indicates that OIDM is helpful in understanding the basic concepts of data mining and the different algorithms. 2) Senior researchers whose feedback suggests that OIDM is helpful in generating preliminary experimental results and conducting data mining research.

In summary, the above feedback suggests that OIDM is a promising toolbox, which could benefit both junior and senior researchers from different perspectives.

6 Conclusion

Recent development in computer network and storage techniques has raised the problem of mining knowledge from large databases generated from both academic research and industrial applications. However, although the extracted knowledge could be very valuable, the efforts in mining are nontrivial, especially for novices in data mining.

In this paper, we have designed a toolbox – OIDM, which supports online interactive data mining, through which the user can get optimal mining results without any programming work. OIDM can interact with the user and help him/her fulfill the learning task. Interactive, complete, and compact are the three goals that guide the system design of OIDM. At the beginning, the user may be ignorant about either the data mining process or the knowledge in the data, and OIDM can take specific steps according to the information gathered from the user and perform the learning task automatically until satisfactory results are found. Compared with other similar toolboxes, OIDM possesses the following unique and useful features. (1) OIDM is a toolbox which can deal with three types of data mining problems (classification, clustering and association mining), whereas most other toolboxes only address one type of data mining problems, and they recommend the best algorithm. (2) The interaction model of OIDM is compact. In other words, it can guide the user to find the right algorithm in only a few steps. (3) OIDM uses a Multi-level Summarization mechanism to present the mining results, which is useful in helping the user understand the results. The evaluation results suggest that, although data mining is a complex task, OIDM can make it simple and flexible through actively involving the user in the mining process.

References

1. Agrawal, R. and Srikant, R. (1994), Fast algorithms for mining association rules, *Proceedings of the 20th VLDB conference*, Santiago, Chile, 1994.
2. Brazdil P., Gama J., and Henery B., (1994), Characterizing the Applicability of Classification Algorithms Using Meta-level learning, *Proc. of* ECML 1994.
3. Brodley C., and Smyth P., (1995), Applying Classification Algorithm in Practice, *Proceedings of the workshop on Applying Machine Learning in Practice at the ICML-95.*
4. Cendrowska, J.. (1987), Prism: An algorithm for inducing modular rules. *International Journal of Man-Machines Studies*, 27: 349-370
5. Clark P., and Niblett T. (1989), The CN2 induction algorithm. Machine Learning, 3, 1989.
6. Craw, Susan (1992), CONSULTANT: Providing Advice for the Machine Learning Toolbox, *Proceedings of the BCS Expert Systems '92 Conference.*
7. Engels R., Lindner G., and Studer R. (1997), A Guided Tour through the Data Mining Jungle, *Proceedings of the 3rd International Conf. on Knowledge Discovery in Database* (1997).
8. Engels, R., Lindner, G., Studer, R. (1998), Providing User Support for Developing Knowledge Discovery Applications; In: S. Wrobel (Ed.) *Themenheft der Kstliche Intelligenz.*
9. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds), *Advances in Knowledge Discovery and Data Mining*, pp. 1-34.
10. Fisher, D.H. (1987), Knowledge acquisition via incremental conceptual clustering, *Machine Learning* 2, pp. 139-172.
11. Graner N., Sharma S., Sleeman D., Rissakis M., Moore C., and Craw S.,(1992), The Machine Learning Toolbox Consultant, *TR AUCS/TR9207*, University of Aberdeen, 1992.
12. Hand D., (1994), Decomposing Statistical Question, *J. of the Royal Statistical Society.*
13. Hellerstein J., Avnur R., Chou A., Hidber C., Olston C., Raman V., Roth T., Haas P., (1999). Interactive Data Analysis: The Control Project, *IEEE Computer*, 32(8), p.51-59, 1999.
14. Heijst V., Terpstra G., Wielinga P., and Shadbolt N., (1992), Using generalised directive models in knowledge acquisition, *Proceedings of EKAW-92*, Springer Verlag.
15. Holte R.C., (1993), Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, 11.
16. Kodratoff Y., Sleeman D., Uszynski M., Causse K., Craw S., (1992), Building a Machine Learning Toolbox, in *Enhancing the Knowledge Engineering Process*, Steels, L., Lepape, B., (Eds.), North-Holland, Elsevier Science Publishers, pp. 81-108.
17. Morales E., (1990), The Machine Learning Toolbox Database, Deliverable 5.8, *Machine Learning Toolbox ESPRIT Project P2154*, 1990.
18. Morik K., Causse K., and Boswell R., (1991), A Common Knowledge Representation Integrating Learning Tools. *Proc. of Workshop on Multi-Strategy Learning*, pp.81-96.

19. Motoda H. (2002), Active Mining, A Spiral Model of Knowledge Discovery, Invited talk of the *2002 IEEE International Conference on Data Mining*, Dec. 9 - 12, Maebashi City, Japan.
20. OIDM (2003), OIDM: A Toolbox to Support Online Interactive Data Mining, http://www.cs.uvm.edu:9180/DMT/index.html
21. Parthasarathy S., and Dwarkadas S. (2002), Shared State for Distributed interactive Data Mining Applications, *International Journal on Distributed and Parallel Databases*.
22. Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*, CA: Morgan Kaufmann.
23. Verdenius F., (1997), Applications of Inductive Learning Techniques: A Survey in the Netherlands, *AI Communications*, 10(1).
24. Ware, Malcolm, Frank, Eibe, Holmes, Geoffrey, Hall, Mark, Witten, Ian H. (2001) Interactive machine learning: letting users build classifiers. *J. of Human Computer Studies* 55(3):
25. White S. and Sleeman D., (1998), Providing Advice on the Acquisition and Reuse of Knowledge Bases in Problem Solving, *Knowledge Acquisition Workshop*, 1998.
26. Wu X., (1995), *Knowledge Acquisition from Databases*, Ablex Publishing Corp., 1995.

A Novel Manufacturing Defect Detection Method Using Data Mining Approach

Wei-Chou Chen, Shian-Shyong Tseng, and Ching-Yao Wang

Department of Computer and Information Science
National Chiao Tung University
Hsinchu 300, Taiwan, R. O. C.
{sirius, sstseng, cywang}@cis.nctu.edu.tw

Abstract. In recent years, the procedure of manufacturing has become more and more complex. In order to meet high expectation on quality target, quick identification of root cause that makes defects is an essential issue. In this paper, we will refer to a typical algorithm of mining association rules and propose a novel interestingness measurement to provide an effective and accurate solution. First, the manufacturing defect detection problem of analyzing the correlation between combinations of machines and the result of defect is defined. Then, we propose an integrated processing procedure RMI (*Root cause Machine Identifier*) to discover the root cause in this problem. Finally, the results of experiments show the accuracy and efficiency of RMI are both well with real manufacturing cases.

1 Introduction

In recent years, the procedure of manufacturing is becoming more and more complex. In order to meet high expectation on yield target, quick identification of root cause that makes defects occur is an essential issue. Therefore, the technologies of process control, statistical analysis and design of experiments are used to establish a solid base for well tuned manufacturing process. However, identification of root cause is still very hard due to the existence of multi-factor and nonlinear interactions in this intermittent problem. Traditionally, the process of identifying root cause for defects is costly. Let's take the semiconductor manufacture industry as an example. With huge amount of semiconductor engineering data stored in database and versatile analytical charting and reporting in production and development, the CIM/MES/EDA systems in the most semiconductor manufacturing companies help users to analyze the collected data in order to achieve the goal of yield enhancement. However, the procedures of semiconductor manufacturing are sophisticated and the collected data among these procedures are thus becoming high-dimensional and huge. In order to deal with the large amount and high-dimensional data, the data mining technologies are thus used to solve such problems. In this paper, we will refer to a typical algorithm of mining association rules and propose a novel interestingness measurement used to evaluate the significance of correlation between combinations of machines and defect to provide an effective and accurate solution. First, the manufacturing defect detection problem, which comprises candidate generation problem and interestingness

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 77–86, 2004.

measurement problem, of analyzing the correlation between combinations of machines and the result of defect is defined. Then, we propose an integrated processing procedure RMI (Root cause Machine Identifier) to discover the root cause in this problem. The RMI procedure consists of three sub-procedures, data preprocessing procedure, candidate generation procedure and interestingness ranking procedure. The purpose of data preprocessing procedure is mainly used to transform raw data into the records to be considered. After that, the candidate generation problem can be efficiently coped with by the candidate generation procedure. Finally, for the interestingness measurement problem, the interestingness of each candidate machineset generated from candidate generation procedure is calculated by our proposed interestingness measurement, and the one with highest value is treated as the root cause, by the interestingness ranking procedure.

2 Related Work

Data mining, which is also referred to as knowledge discovery in database, means a process of the nontrivial extraction of implicit, previously unknown and potentially useful information from databases [7][13]. According to the classes of knowledge derived, the mining approaches may be classified as finding association rules [1-4][8][18-20][28], classification rules [6][22][23][27], clustering rules [9][15][17][29], etc [10][13]. Among them, finding association rules in transaction databases is most commonly seen in data mining.

Conceptually, an association rule indicates that the occurrence of a set of items (or itemset) in a transaction would imply the occurrence of other items in the same transaction [1]. The processing procedure of mining association rules can be typically decomposed into two tasks [2]: (a) discover the itemsets satisfying the user-specified minimum support from a given dataset, i.e. finding frequent itemsets, and (b) generate strong rules satisfying the user-specified minimum confidence from the frequent itemsets, i.e. generating association rules. The task (a) is used to obtain the statistically significant itemsets, and the task (b) is used to obtain the interesting rules.

The major challenge is how to reduce the search space and decrease the computation time in task (a). Apriori algorithm, introduced in [2], utilizes a level-wise candidate generation approach to dramatically prune the itemsets without satisfying the user-specified minimum support, and then derives the association rules which satisfy the user-specified minimum confidence from the remaining ones (i.e. from frequent itemsets). Although significant itemsets can be efficiently discovered by Apriori algorithm, only few of them may be interesting for users. Therefore, to design a useful interestingness measurement is becoming an important research issue [3][7][13][26]. Confidence, the most typical interestingness measurement for association rule mining, measures the conditional probability of events associated with a particular rule. There are many efforts thus attempt to propose other effective interestingness measurements [3][11][14][21][25][26]. In [21], Piatetsky-Shaprio proposed a domain-independent interestingness measurement Formula (**1**) to evaluate the interestingness of discovered rules:

$$\phi = \frac{|A \,\&\, B| - |A||B|/N}{\sqrt{|A||B|(1-|A|/N)(1-|B|/N)}} \tag{1}$$

Under a form of rule A→B, let N be the total number of tuples in the database, |A| be the number of tuples that include the antecedent A, |B| be the number of tuples that include the consequent B, and |A&B| be the number of tuples include A and B. The range of this interestingness measurement is between –0.25 and 0.25.

3 Problem Formulation

In a general manufacturing process, the manufacture of a product may require a multi-stage procedure. In each stage, it may have more than one machine doing the same task. Products may therefore be passed through different machines in the same stage. We can thus define our problem as follows. Suppose a shipment consists of k identical products $\{p_1, p_2, \ldots, p_k\}$ and each product should pass through l stages $<s_1, s_2, \ldots, s_l>$ sequentially. A stage may contain one or more identical machines which do the same task. Therefore, let $M = \{m_{ij} \mid 1 \le i \le l,\ 1 \le j \le \sigma_i\}$ be a set of machines in the manufacturing procedure with l stages, where m_{ij} denotes the jth machine in ith stage and σ_i is the number of identical machines in the ith stage. A *manufacturing process relation* $r = \{t_1, t_2, \ldots, t_k\}$ based on a schema $(PID, S_1, S_2, \ldots, S_l, D)$ can be used to record the sequentially processing procedure for each product, including the machine in each stage and its finally testing result, where PID is an identification attribute used to uniquely label the products, $S_i = <m_{ij}, t_i>$, $1 \le i \le l$, is a context attribute used to record the pair of the processing machine in ith stage and the timestamp after this stage, and D is a class attribute used to represent whether the product p is a defect or not. For Example, Table 1 shows a manufacturing process relation used to record a three-stage procedure for a shipment consisting of five products. The tuple with PID = 1 shows that the Product 1 passes through stage 1 by $<m_{11}, 1>$, stage 2 by $<m_{21}, 3>$, and then stage 3 by $<m_{31}, 10>$, and its testing result is defective. The other tuples have similar meaning.

Table 1. An example of a three-stage procedure for five products

PID	S_1	S_2	S_3	*D*
1	m_{11}, *1*	m_{21}, *3*	m_{31}, *10*	*1*
2	m_{12}, 5	m_{21}, 8	m_{32}, 12	0
3	m_{11}, 2	m_{22}, 7	m_{31}, 13	1
4	m_{11}, 4	m_{22}, 6	m_{32}, 14	1
5	m_{12}, 7	m_{21}, 11	m_{31}, 15	0

3.1 Decomposition of Manufacturing Defect Detection Problem

The goal of manufacturing defect detection problem is used to discover the root cause machineset from the given data. The key idea of this problem is to find out the machineset which is *rather relative* to defect. The first challenge is how to design an effective interestingness measurement to evaluate the correlation between machinesets and defect. Moreover, the computation costs of enumerating all combinations of the machines and then evaluating these machinesets are relatively enormous. The second challenge is how to develop a pruning strategy to remove the machinesets which have not enough evidence to be a root cause to reduce the search space and decrease the computation time. As the results, this problem can be decomposed into two sub-problems, *candidate generation problem* and *interestingness measurement problem*. The candidate generation problem focuses on how to generate the candidate machineset, and the interestingness measurement problem focuses on how to identify the root cause machineset.

In the candidate generation problem, we refer to typical algorithms in mining association rules to generate possible machinesets for root cause. Just like frequent itemsets generation in association rules mining, a level-wise processing procedure is required to generate the combinations of machines above a certain threshold, which are called *candidate machineset*. Instead of counting fractional transaction support of itemsets in association rules mining, counting *defect coverage* of machinesets is introduced in this procedure. The defect coverage of a machineset denotes a percentage of the number of defects that the target machineset involves in the products.

Example 2: For each machine in Table 1, the defect coverage is shown in Table 2. The first tuple shows that three products, p_1, p_3 and p_4, pass through m_{11} and all of them are defective. The defect coverage of m_{11} is thus 60%.

Table 2. The defect coverage for each machine in Table 1

Machine	*Involved Products*	*Defect Coverage*
m_{11}	p_1, p_3, p_4	*60%*
m_{12}	p_2, p_5	*0*
m_{21}	p_1, p_2, p_5	*20%*
m_{22}	p_3, p_4	*40%*
m_{31}	p_1, p_3, p_5	*40%*
m_{32}	p_2, p_4	*20%*

Therefore, given a user-specific *minimum defect coverage*, in the first pass the candidate generation procedure will calculate the defect coverage of individual machines and retain ones of them that are more than user-specific minimum defect coverage as candidate 1-machinesets (a machineset consists of one machine). In the second pass, we generate the machineset consisting of 2 machines by joining the candidate 1-machinesets and then retain the 2-machinesets that satisfy the minimum defect coverage constraint. In subsequent passes, the candidate machinesets found in the previous pass are treated as a seed set for this pass. This process continues until no new candidate machinesets are generated.

Example 3, continuing Example 2, suppose the user-specific minimum defect coverage is 40%, and m_{12}, m_{21} and m_{32} are then removed since their defect coverages are all less than 40%. All candidate 1-machinesets are thus shown in Table 3. The first tuple shows that the defect coverage of m_{11} is 60%, and the corresponding defective products are p_1, p_3 and p_4

Table 3. The defect coverage and the information of defective products for each candidate 1-machineset

Machine	*Defect Coverage*	*Defective Products*
m_{11}	60%	p_1, p_3, p_4
m_{22}	40%	p_3, p_4
m_{31}	40%	p_1, p_3

Next, 2-machinesets, $\{m_{11}, m_{22}\}$, $\{m_{11}, m_{31}\}$ and $\{m_{22}, m_{31}\}$, are generated by joining the candidate 1-machinesets in Table 3 to next level. The defect coverages of these three 2-machinesets and their related information of defective products can be calculated by utilizing the information of defective products in Table 3. For example, the defect coverage and the information of defective products for $\{m_{11}, m_{22}\}$ are 40% and $\{p_3, p_4\}$. The results are shown in Table 4.

Table 4. The defect coverage and the information of defective products for 2-machinesets $\{m_{11}, m_{22}\}$, $\{m_{11}, m_{31}\}$ and $\{m_{22}, m_{31}\}$

Machine	*Defect Coverage*	*Defective Products*
m_{11}, m_{22}	40%	p_3, p_4
m_{11}, m_{31}	40%	p_1, p_3
m_{22}, m_{31}	20%	p_3

$\{m_{22}, m_{31}\}$ is removed since its defect coverage is less than the user-specific minimum defect coverage. The candidate 2-machinesets are thus shown in Table 5.

Table 5. The defect coverage and the information of defective products for each candidate 2-machineset

Machine	*Defect Coverage*	*Defective Products*
m_{11}, m_{22}	40%	p_3, p_4
m_{11}, m_{31}	40%	p_1, p_3

In the third level, the only one 3-machineset $\{m_{11}, m_{22}, m_{31}\}$ is generated by joining the candidate 2-machinesets in Table 5. However, since $\{m_{22}, m_{31}\}$ is not included in the set of candidate 2-machinesets, it is removed according to above-mentioned Apriori property. Consequently, all candidate machinesets are generated as shown in Table 6.

Table 6. The defect coverage and the information of defective products for all candidate machinesets

Machine	*Defect Coverage*	*Defective Products*
m_{11}	60%	p_1, p_3, p_4
m_{22}	40%	p_3, p_4
m_{31}	40%	p_1, p_3
m_{11}, m_{22}	40%	p_3, p_4
m_{11}, m_{31}	40%	p_1, p_3

Although that a candidate machineset has high defect coverage is statistically significant, it may not have high possibility to be root cause. In [21], Piatetsky-

Shaprio proposed a domain-independent interestingness measurement as shown in Formula (**1**) to evaluate the discovered rules. Such equation indicates the degree of "when antecedent *A* appears, consequent *B* appears, too". Therefore, if we regard *A* as a certain machineset and *B* as the defect, then by the equation the degree of correlation between a machineset and the defect can be calculated. However, it does not consider characteristics of manufacturing process; we thus introduce an additional measure function called *continuity* for enhancement.

The *continuity* is a measure used to evaluate the degree of continuity of defects in the products in which a target machineset is involved. The higher value of continuity means that the frequency of defect occurrence in the involved products is higher and the corresponding machineset have higher possibility to be the root cause. This enhancement is due to the observation that the defects in the involved products for root cause machineset often occur continuously in the real world. It can be calculated by the reciprocal of the average distance of each pair of neighboring defects in the involved product sequence as Formula (**2**)

$$\begin{cases} Continutiy = 0 & if|X| \le 1 \\ Continuity = \dfrac{1}{\sum_{i=1}^{i=|X|-1} d(\alpha(x_i),\alpha(x_{i+1})) \Big/ |X|-1} & if|X| > 1 \end{cases} \tag{2}$$

,

where $X = (x_1, x_2, \ldots)$ denotes a sequence consisting of the defective products in the involved product sequence $P = (p_1, p_2, \ldots)$ for a machineset (i.e., X is a subsequence of P), $|X|$ denotes the number of defective products, $\alpha(x_i)$ denotes the defective product x_i is the k-th product in the involved product sequence if $\alpha(x_i) = \mathrm{k}$, and $d(\alpha(x_i),\alpha(x_{i+1}))$ is the distance of $\alpha(x_i)$ and $\alpha(x_{i+1})$ and it can be easily calculated by $\alpha(x_{i+1})-\alpha(x_i)$.

Example 4: Table 7 shows the involved product sequences, defective product sequences and calculated continuities for two machines m_1 and m_2, respectively. The first tuple shows that the involved product sequence is $(p_1, p_3, p_4, p_5, p_6)$ and the defective product sequence is (p_1, p_3, p_5), and therefore the continuity of m_1 can be calculated by $\dfrac{1}{(d(\alpha(p_1),\alpha(p_3)) + d(\alpha(p_3),\alpha(p_5)))/(3-1)} = \dfrac{1}{(1+2)/2} = 0.67$. The continuity of m_2 is 0 since the number of the defects is 0.

Table 7. The calculated continuities for two machines m1 and m2

Machine	*Involved Product Sequence*	*Defective Product Sequence*	*Continuity*
m_1	$(p_1, p_3, p_4, p_5, p_6)$	(p_1, p_3, p_5)	*0.67*
m_2	(p_2, p_4, p_6)		*0*

We can easily extend the interestingness measurement ϕ by multiplying continuity and ϕ together as Formula (**3**):

$$\phi' = \frac{|A \,\&\, B| - |A||B|/N}{\sqrt{|A||B|(1-|A|/N)(1-|B|/N)}} * Continuity \tag{3}$$

Consequently, the extended interestingness measurement ϕ' is used to each of candidate machinesets discovered in the candidate generation procedure, and then sorting these candidate machinesets by the calculated values. The machineset with highest interestingness value is treated as the root cause machineset.

Example 5: Continuing Example 4, the corresponding ϕ' for each candidate machinesets can be calculated by multiplying continuity and ϕ together. The results are shown in Table 8. Since the candidate machineset m_{11} has highest interestingness value, it will be treated as the root cause machineset.

Table 8. The calculated ф' for the candidate machinesets in Table 6

Machine	ϕ	*Continuity*	*Interestingness* ϕ'
m_{11}	*1*	*1*	*1*
m_{22}	*0.67*	*1*	*0.67*
m_{31}	*0.167*	*1*	*0.167*
m_{11}, m_{22}	*0.67*	*1*	*0.67*
m_{11}, m_{31}	*0.67*	*1*	*0.67*

3.2 An Integrated Processing Procedure: Root cause Machine Identifier (RMI)

After the *manufacturing defect detection problem* is generated according to the above method, an integrated processing procedure called *Root cause Machine Identifier* (RMI) is proposed to solve this problem. At first, The RMI procedure will get "*pure*" data from the raw data in the operational database by a *data preprocessing* procedure. After that, by the candidate generation procedure, all candidate machinesets (i.e. the machinesets satisfying user-specified minimum defect coverage) can be generated from the preprocessed records. Finally, by the interestingness ranking procedure, the interestingness of each candidate machineset is calculated by the proposed interestingness measurement ϕ', and the one with highest value is treated as the root cause machineset.

4 Experimental Results

In our experiment environment, we implement RMI procedure in Java language on a Pentium-IV 2.4G processor desktop with 512MB RAM, and nine real datasets provided by *Taiwan Semiconductor Manufacturing Company* (TSMC) are used to evaluate the accuracy. Since root cause machinesets of these nine real datasets have been identified, this information can be used to evaluate the accuracy of RMI.

Table 9 shows the related information for the nine real datasets after the data preprocessing procedure. For example, the first tuple shows that the first dataset called Case 1 has 152 products and each one passes through 1318 stages to be finished, and there are 2726 machines in that.

Table 9. Related information for the nine real datasets

Case name	*Data size (Products*Stages)*	*Number of machines*
Case 1	152*1318	2726
Case 2	277*1704	4370
Case 3	239*1436	2004
Case 4	126*1736	4437
Case 5	139*1676	4410
Case 6	114*1250	3485
Case 7	53*1176	2414
Case 8	484*1420	3381
Case 9	106*1266	2618

In the accuracy evaluation, we set the minimum defect coverage ranging from 0.4 to 0.6. The results are shown in Table 10. It shows the rank of the root cause machineset among all candidate machinesets generated by RMI. Note that "X" means can't discover the root cause machineset.

Table 10. Accuracy result of RMI for the nine datasets

Case Name	*Rank (min_def_cov=0.4)*	*Rank (min_def_cov=0.5)*	*Rank (min_def_cov=0.6)*
Case 1	10	5	2
Case 2	1	X	X
Case 3	16	7	3
Case 4	1	1	1
Case 5	1	1	X
Case 6	4	4	4
Case 7	167	101	62
Case 8	1	1	X
Case 9	2	2	X

By RMI, the root cause machineset in most of datasets can be ranked in the top five with suitable minimum defect coverage, except the Case 7. This major reason is that only 53 products are given in Case 7, and therefore the root cause machineset is not significant comparing with others.

5 Conclusion

Identification of root cause for defects in manufacturing can not only reduce the manufacturing cost, but also improve the performance of a manufactory. However, traditional methodologies of identifying the root cause are restricted or depend on experiences and expertise. In this paper, we refer to typical association rule mining algorithms to propose an integrated processing procedure called RMI to provide an effective and accurate solution. By the level-wise candidate generation procedure, the RMI can efficiently generate the candidate machinesets for root cause, the interestingness of each generated candidate machineset is then calculated by our proposed interestingness measurement, and the one with highest value is consequently treated as the root cause by the interestingness ranking procedure. From the experiment results in the real datasets, the accuracy of our proposed interestingness measurement always outperforms other ones.

Acknowledgement. This work was partially supported by MOE Program for Promoting Academic Excellence of Universities under grant number 89-E-FA04-1-4, High Confidence Information Systems.

References

[1] Agrawal, R., Imielinski, T. and Swami, A.: Mining association rules between sets of items in large database. Proc. ACM SIGMOD Conference (1993) 207-216.

[2] Agrawl, R. and Srikant, R.: Fast Algorithm for Mining Association rules. Proc. ACM VLDB Conference (1994) 487-499.

[3] Brin, S., Motwani, R. and Silverstein, C.: Beyond Market Basket: Generalizing Association Rules to Correlations. Proceeding of ACM SIGMOD Conference (1997) 265-276.

[4] Brin, S., Motwani, R., Ullman, J. D. and Tsur S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data. Proc. ACM SIGMOD Conference (1997) 255-264.

[5] Catledge, L. D. and Pitkow, J. E.: Characterizing Browsing Strategies in the World Wide Web. Proc. Third WWW Conference, Apr. 1995.

[6] Cheeseman, P. and Stutz., J.: Bayesian Classification (AutoClass): Theory and Results. In Fayyad, U.M., G., Smyth, P. and Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press (1996) 153-180,.

[7] Chen, M. S., Han, J. and Yu, P. S.: Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.6 (1996).

[8] Cheung, D. W., Han, J., Ng, V. T. and Wong, C. Y.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Approach, Proc. IEEE International Conference on Data Engineering (1996) 106-114

[9] Ester, M., Kriegel, H. O. and Xu, X.: Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification, Proc. Fourth International Symp. Large Spatial Databases, 67-82.

[10] Faloutsos, C., Ranganathan, M. and Manolopoulos, Y.: Fast Subsequence Matching in Time-Series Databases. Proc. ACM SIGMOD Conference (1994). 419-429.

[11] Freitas A. A.: On Rule Interestingness Measures, Knowledge-Based System. (1999) 309-315.

[12] Gardner, M. and Bieker, J.: Mining Solves Tough Semiconductor Manufacturing Problems. Proc. ACM KDD Conference (2000).

[13] Han, J. and Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001).

[14] Hilderman, R. J. and Hamilton, H. J.: Heuristic Measures of Interestingness. Principles of Data Mining and Knowledge Discovery (1999) 232-241.

[15] Kaufuman, L. and Rousseeuw, P. J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wisley & Sons (1990).

[16] Mieno, F., Santo, T., Shibuya, Y., Odagiri, K., Tsuda, H. and Take, R.: Yield Improvement Using Data Mining System. Proc. IEEE Semiconductor Manufacturing Conference (1999).

[17] Ng, R. and Han, J.: Efficient and Effective Clustering Method for Spatial Data Mining. Proc. ACM VLDB Conference (1994) 144-155

[18] Park, J. S., Chen, M. S. Yu, and P. S.: An effective hash-based algorithm for mining association rules. Proc. ACM SIGMOD Conference (1995) 175-186.

[19] Park, J. S., Chen, M. S. and Yu, P. S.: Mining Association Rules with Adjustable Accuracy. IBM Research Report (1995).

[20] Park, J. S., Chen, M. S. and Yu, P. S.: Efficient Parallel Data Mining for Association Rules. Proc. ACM CIKM Conference (1995) 175-186.

[21] Piatestsky-Shaprioc G., Discovery, Analysis and Presentation of Strong Rules. In Piatetsky-Shapiro, G and, Frawley, W. J. (eds.., Knowledge Discovery in Databases, AAAI Press (1991) 229-247.

[22] Quinlanc J. R.: Induction of Decision Trees. Machine Learning, Vol. 1 (1986).81-106.

[23] Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Press (1993).

[24] Raghavan V.: Application of Decision Trees for Integrated Circuit Yield Improvement. Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conference & Workshop (2002).

[25] Silberschatz A. and Tuzhilin, A.: What Makes Patterns Interesting in Knowledge Discovery Systems. IEEE Transaction on Knowledge and Data Engineering (1996).

[26] Tan, P. N. and Kumar, V.: Interestingness Measures for Association Patterns: A Perspective, Proc. KDD'2000 Workshop on Postprocessing in Machine Learning and Data Mining (2000).

[27] Weiss, S. M. and Kulikowski, C. A.: Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets. Machine Learning and Expert Systems. Morgan Kaufman Press (1991).

[28] Wur, S. Y. and Leu, Y.: An Effective Boolean Algorithm for Mining Association Rules in Large Databases. Proc. International Conference on Database Systems for Advanced Applications (1999).

[29] Zhang T., Ramakrishnan R. and Livny M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases, Proc. ACM SIGMOD Conference *(1996)*. 103-114.

Neural Representation of a Solar Collector with Statistical Optimization of the Training Set

Luis E. Zárate[1], Elizabeth Marques Duarte Pereira[2], João Paulo Domingos Silva[1], Renato Vimeiro[1], and Antônia Sônia Cardoso Diniz[3]

[1] Applied Computational Intelligence Laboratory (LICAP)
[2] Energy Researches Group (GREEN)
[3] Energy Company of Minas Gerais (CEMIG)
Pontifical Catholic University of Minas Gerais (PUC)
500 Dom Jośe Gaspar Ave., Coração Eucarístico
Belo Horizonte, MG, Brazil, 30535-610
{zarate, green}@pucminas.br

Abstract. Alternative ways of energy producing are essential in a reality where natural resources have been scarce and solar collectors are one of these ways. However the mathematical modeling of solar collectors involves parameters that may lead to nonlinear equations. Due to their facility of solving nonlinear problems, ANN (i.e. Artificial Neural Networks) are presented here, as an alternative to represent these solar collectors with several advantages on other techniques of modeling, like linear regression. Techniques for selecting representative training sets are also discussed and presented in this paper.

1 Introduction

Due to the hydrographic basins situation around the world associated with a constant population increasing, a new reality can be noticed, where alternative ways of energy producing have essential importance. Solar energy systems are one of these ways.

Solar energy systems, specifically water heaters, have considerable importance as substitutes of traditional electrical systems. In Figure 1, an example of water heater, called thermosiphon system, is schematically represented. The solar collector (collector plate) is the most important component in a thermosiphon system.

The performance of thermosiphon systems has been investigated, both analytically and experimentally, by numerous researches [1-3]. The formula used to calculate the solar collector efficiency is the following:

$$\eta = \frac{\dot{m}\, c_p (T_{out} - T_{in})}{G A_{extern}} . \tag{1}$$

In the formula above, η is the thermal efficiency, $\dot{m}$, the flow rate, c_p, the heat capacity of water, T_{out}, the output temperature of water, T_{in}, the input temperature of water, G, the solar irradiance and A_{extern}, the area of the collector.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 87-96, 2004.

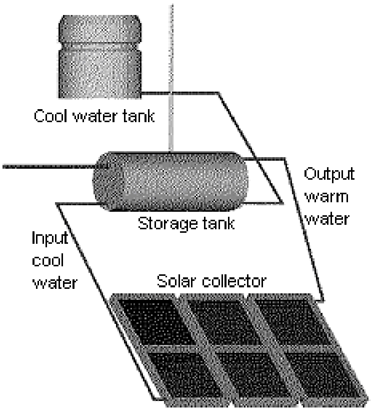


Fig. 1. Schematic diagram of a thermosiphon system

Mathematical models [1-5] have already been presented as a way of calculating the efficiency of solar collectors; however the non-linearity nature of those models makes their application discouraging. Linear regression [3] has been proposed as a way of modeling solar collectors, instead of those complex mathematical models. But linear regression may introduce significant errors when used with that purpose, due to its limitation of working better only with linear correlated values.

In the last years, ANN (i.e. Artificial Neural Networks) have been proposed as a powerful computational tool. Some researches [5] and [10] have already discussed their use in the representation of thermosiphon systems. ANN have several advantages on other techniques, including their performance when dealing with nonlinear problems, their capacity of learning and generalizing, the low time of processing that can be reached when trained nets are in operation etc.

This paper is organized in four sections. In the second one, solar collectors are physically described. In the third section, the use of ANN in the representation of solar collectors and statistical techniques of training sets selection are discussed. And in the fourth section, conclusions are presented.

2 Physical Description of the Solar Collector

The working principles of thermosiphon systems are based on thermodynamic laws [6]. In those systems, water circulates through the solar collector due to the natural density difference between cooler water in the storage tank and warmer water in the collector. Although they demand larger cares in their installation, thermosiphon systems are of extreme reliability and lower maintenance. Their application is restricted to residential, small commercial and industrial installations.

Solar irradiance reaches the collectors, which heat up water inside them, decreasing the density of heated up water. Thus cooler and denser water forces warm water to the storage tank. The constant water flow that happens between the storage tank and the collector results in a natural circulation, called □thermosiphon effect□

In a very similar way that it happens during the day, thermosiphon tends to realize the opposite process during the night, cooling water. In order to avoid this opposite effect, a minimum height between the collector and the storage tank is expected to be established. This height is typically 30cm.

3 Neural Representation of the Solar Collector

Multi-layer ANN have been used in this work. The values of entries are presented to the hidden layer and satisfactory answers are expected to be obtained from the output layer. The most suitable number of neurons in the hidden layer is still a non-solved problem, although researches discuss some approaches. In [7], the suggested number of hidden neurons is *2n+1*, where *n* is the number of entries. In the other hand, the number of output neurons equals the number of expected answers from the net.

Input water temperature (T_{in}), solar irradiance (G) and ambient temperature (T_{amb}) are variables used as entries to the ANN. The output water temperature (T_{out}) is the wanted output from the net. In this work, ANN represent the thermosiphon system according to the following formula

$$f(T_{in}, T_{amb}, G) \xrightarrow{ANN} T_{out} \,. \tag{2}$$

The structure of the ANN in this work is schematically represented as shown in the Figure 2. The net contains seven hidden neurons (i.e. *2n+1*) and one neuron in the output layer, from which the output water temperature is obtained.

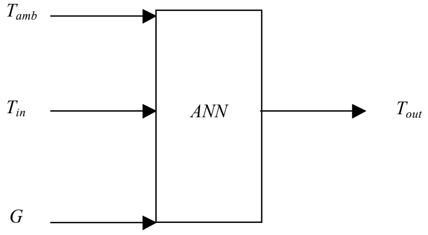


Fig. 2. Schematic diagram of the net used in this work

Supervised learning has been adopted to train the net, specifically, the widely used algorithm known as backpropagation. Nonlinear sigmoid function has been chosen, in this work, as the axon transfer function:

$$f = \frac{1}{1 + \exp^{-\sum \textit{Entries} \text{ x Weigths}}} \tag{3}$$

Backpropagation algorithm uses a training set, from where inputs and wanted outputs may be extracted. For each parameter of the chosen training set, the weights of the net are adjusted in order to minimize errors obtained in the outputs values.

3.1 Collecting Data from the Solar Collector

Data have been collected by means of experiments under specific standards [9], during three days in different ambient situations, and refer to a typical solar collector. Figure 3 shows the relation between the output water temperature (T_{out}) and the hours of collecting (*hours*). It can be observed that collected data cover several situations.

In order to verify the non-linearity of the data, Figures 4, 5 and 6 have been built. However those graphics demonstrate that there is linearity in the collected data. Although linear regression gives a valid approach, this work tries to demonstrate that ANN are capable of estimating output temperatures with more precision.

The total number of collected data equals 633 (a sample of those data is shown in Table I.1, in the appendix). A subset containing 30 data has been removed from the initial set, with the purpose of been used in the validation process of the net, as shown in the next sections. 603 data still remain in the training set.

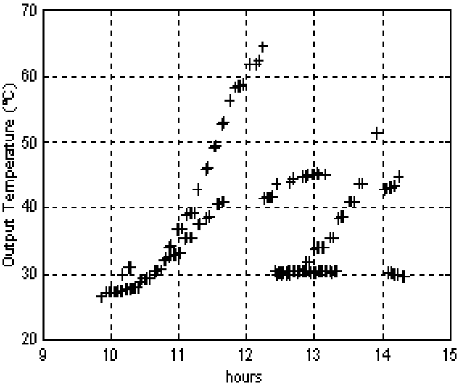


Fig. 3. Collected water output temperatures

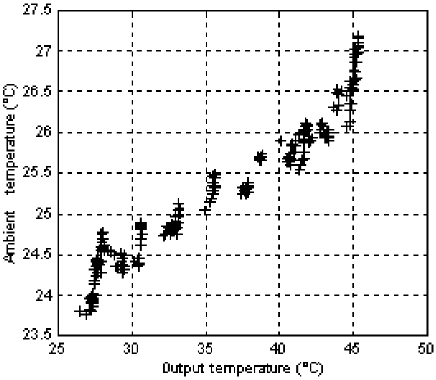


Fig. 4. *Tamb x Tout*

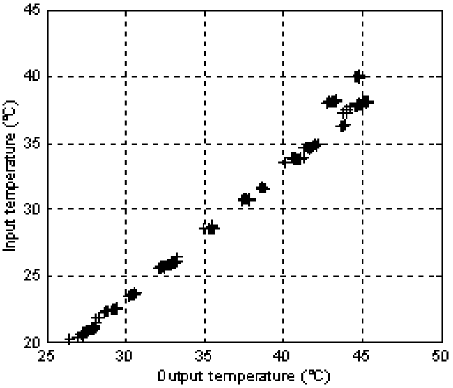


Fig. 5. *Tin x Tout*

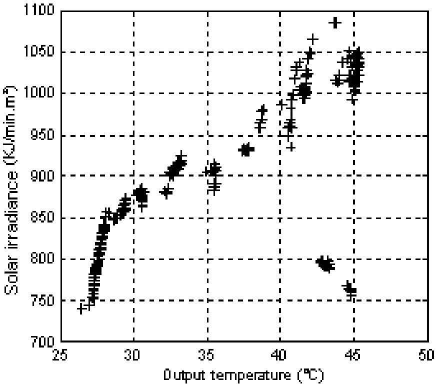


Fig. 1. *Solar irradiance X Tout*

3.2 Preparing Data for Training

Pre-processing input data is a process of considerable importance for the performance of ANN. In this work, the following procedure has been applied to collected data, before the use of them in the net structure:

1. Data values have been normalized in order to be within the interval [0.2, 0.8].
2. The following formulas have been used in the normalization process:

$$f^{a}(\mathrm{L}o) = \mathrm{L}n = (\mathrm{L}o - \mathrm{L}min) / (\mathrm{L}\max - \mathrm{L}\min) \tag{4}$$

$$f^{b}(\mathrm{L}n) = \mathrm{L}o = \mathrm{L}n * \mathrm{L}\max + (1 - \mathrm{L}n) * \mathrm{L}min \tag{5}$$

The formulas above must be applied to each variable of the training set (e.g. T_{amb}, T_{in}, G), normalizing all their values.

3. *Lmin* and *Lmax* have been computed as follows

$$L_{\min} = L_{\sup} - (N_s / (N_i - N_s)) * (L_{\inf} - L_{\sup}) \tag{6}$$

$$L_{\max} = ((L_{\inf} - L_{\sup}) / (N_i - N_s)) + L_{\min} \tag{7}$$

where L_{sup} is the maximum value of that variable, L_{inf} is its minimum value, N_i and N_s are the limits for the normalization (in this case, $N_i = 0.2$ and $N_s = 0.8$).

3.3 Selecting Data for Training

The actual training set contains 603 data, but a reduction in its size may be beneficial, decreasing the time spent in the training process and also maintaining the capacity of generalization of the trained net. The literature [8] suggests some techniques to build small better-defined training sets. Even if smaller, trainings sets must cover all the possible situations that may happen in the problem or, at least, a major part of them.

The following formula has been borrowed from statistics area and has been used in this work to calculate suitable sizes of training sets

$$n = \left(\frac{z}{e}\right)^2 * (f * (1 - f)) \tag{8}$$

where n is the size of the set, z is the reliance level, e is the error around the average and f is the population proportion.

Fixing z in 90%(z = 1.645), f in 0.5 and varying e from 0.04°C to 0.1°C, several sizes of training sets have been calculated (Table 1). It is extremely important to emphasize that errors values considered here are not the maximum errors of the net.

Table 1. Calculated set sizes

Value *e*	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Set size	423	271	188	139	106	84	68

The following procedure has been adopted to build trainings sets with each one of the calculated sizes, selecting the necessary quantities of data among the 603 data:

1. For each variable of the net, the maximum and the minimum values found in the set have been selected.
2. The operation point of the problem (i.e. the most representative parameter in the set) has been chosen and added to the new training set.
3. The remaining elements have been randomly selected, with the purpose of reaching the calculated size for each training set.

Eight training sets of different sizes (seven presented in Table 1 and the original one of 603 data) have been built. The most representative of them must be chosen.

3.4 Training Process

Several nets have been trained with each one of the different training sets. The net has been configured with 7 neurons in the hidden layer, learning rate equivalent to 0.08, maximum error of 0.05 and all the initial weights on zero. The necessary number of iterations to train each of the nets is presented in Figure 7.

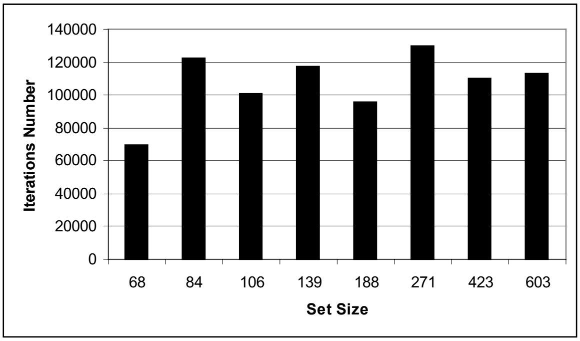


Fig. 7. Number of iterations in the training of each net

Table 2 presents the average errors obtained in the validation process of each trained net. For the validation process, the 30 previously separated data has been used.

Table 2. Average error of the validation process of each trained net

Size set	68	84	106	139	188	271	423	603
Error	0.8544	0.6005	0.7367	0.5616	0.7328	0.5831	0.6674	0.6213

The training set composed by 84 data has been chosen, because the net trained with it maintains a satisfactory capacity of generalization (better than the one trained with 603 data), despite of the fact of spending a larger number of iterations.

In order to verify the performance of the chosen net, another net has been trained with the training set of size 84 and the same initial configuration. However, the maximum tolerable error has been changed and now it is equivalent to 1□C (0.016 in normalized values), and the weights have been initialized with random values instead of a constant zero value. Figure 8 shows the results of that new training.

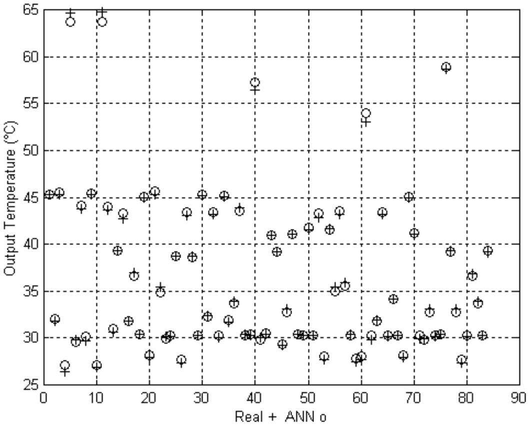


Fig. 8. *Real (o)* and *ANN (+)* output temperatures

3.5 Validation Process

Table I.3 (in the appendix) shows the data set used to validate the trained net and also shows the output of the net trained with 84 data and its errors. A comparison between validation processes involving nets trained with 84 and 603 data is shown in Table 3. In Figure 9, the results of the validation process are graphically shown.

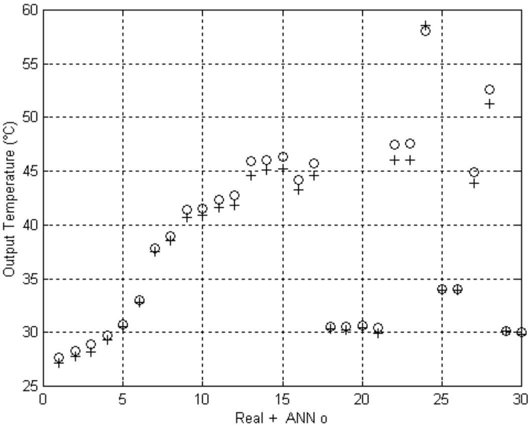


Fig. 9. Results of the validation process of the net trained with 84 data

Table 3. Comparison of errors obtained in the process of validation for different nets

	Error (□C) with 84 sets	Error (□C) with 603 sets
Minimum	0.043265	0.02185
Maximum	1.475292	0.70706
Average	0.625579	0.27365

Error values obtained in the validation process of the net trained with 84 data are greater than the ones of the net trained with 603 data; however, its average error is lower than 1□C, as recommended by INMETRO (National Institute of Metrology and Industrial Quality - Brazil). Above all, the number of iterations in the training with 84 and 603 data, with the last configuration, are, respectively, 412800 and 7.7 million.

3.6 Verification of Results

For the analysis by means of linear regression, Equation (9) has been used:

$$\eta = F_R (\tau\alpha)_e - F_R U_L \frac{(T_{in} - T_{amb})}{G} \tag{9}$$

$F_R(\tau\alpha)_e$ equals 66.662 and F_RU_L, 809.89. F_R corresponds to collector heat removal factor, $(\tau\alpha)_e$, to transmittance absorptance product and U_L, to collector overall loss coefficient. T_{in} is the input water temperature, T_{amb}, the ambient temperature and G, the solar irradiance. Equation (9) calculates efficiency when linear regression is used.

The solar collector efficiency can be calculated by means of ANN since the output water temperature is obtained. Once trained, a net can estimate values of output water temperature for parameters not present in the training process and, consequently, the efficiency can be calculated. Table I.2 (appendix) shows a sample of real efficiency values and their respective estimated values.

For the 84 data (containing T_{in}, T_{amb}, G, T_{out}) used by the net in its training, the real efficiency values have been calculated using Equation (1). Linear regression and the net (trained with those 84 data) have been used to calculate efficiency too, with their already mentioned respective manners of doing that. With real and estimated values of efficiency, errors of each one of the both techniques can be calculated. Table 4 shows a comparison between those error values of ANN and linear regression.

Table 4. Error values in the calculation of solar collector efficiency

Efficiency calculation	Errors of ANN (%)	Errors of LR (%)
Minimum	0.0003	0.0671
Maximum	8.9414	10.5667
Average	2.2710	2.1041
Standard deviation	2.0374	2.0823

Even though the average error obtained in the linear regression technique is lower than the one obtained in ANN technique, the standard deviation value indicates that the error values of ANN are less dispersed than the error values of linear regression. It can be also noticed that a smaller better-defined training set has not prejudiced the generalization capacity of the net, once its errors are satisfactory.

4 Conclusions

A possible use of Artificial Neural Networks has been proposed in this work for the representation of a solar collector, alternatively to other techniques already used with the same purpose. Figure 8 graphically represents the performance of the trained net in the training process, and Table 3 shows its errors in the validation process, with 30 parameters not seen during the training. Efficiency values, calculated using outputs of the trained net, are shown in Table 4.

Analyzing Tables 3 and 4, it can be noticed that the size reduction of the training set, by means of statistical techniques, has maintained the generalization capacity and low error values of the net, besides its contribution in decreasing the time spent in the training. Table 4 also shows a comparison between ANN and linear regression in the calculation of solar collector efficiency. Linear regression errors are more dispersed than ANN ones, indicating that ANN are an advantageous alternative. Thus, ANN have been considered a satisfactory alternative in the proposed problem.

Since the size reduction of the training set has been profitable and the use of ANN instead linear regression presents advantages, future works in the same area are being planned. One of them includes the use of other techniques in the selection of data for the training set. In other future works, ANN will be used in the modeling of many kinds of solar collectors, each one with its own geometrical parameters.

Acknowledgements. This project is financially supported by CEMIG (Energy Company of Minas Gerais).

References

1. G.L. Morrison, and D.B.J. Ranatunga, Transient response of thermosiphon solar collectors. Solar Energy 24, 1980, 191.
2. B.J. Huang, Similarity theory of solar water heater with natural circulation. Solar Energy 25, 1984, 105.
3. A.I. Kudish, P. Santaura, and P. Beaufort, Direct measurement and analysis of thermosiphon flow. Solar Energy, Vol. 35, N□2, 167-173
4. S.A. Kalogirou, Thermosiphon solar domestic water heating systems: long term performance prediction using ANN, Solar Energy, Vol. 69, N□2, 2000, 167-174.
5. S.A. Kalogirou, S. Panteliou, A. Dentsoras, Modeling solar domestic water heating systems using ANN, Solar Energy, Vol. 68, N□6, 1999, 335-342.
6. J.A. Duffie, and W.A. Beckman, Solar Engineering of Thermal Processes, (2a. Ed. John Wyley&Sons, Inc., USA 1999).
7. Z.L. Kov□cs, □Redes Neurais Artificiais□, (Edi□□o acad□mica S□o Paulo, Cap 5, pp.75-76. S□o Paulo. Brasil, 1996).
8. F. R. Bittencout and L.E. Z□rate, Controle da Lamina□□o em Redes Neurais, com Capacidade de Generaliza□□o, e L□gica Nebulosa via Fatores de Sensibilidade, CBA, Natal, RN, 2002.
9. Ashrae 93-86 Ra 91. Methods of Testing to Determine the Thermal Performance of Solar Collectors, American Society of Heating, Refrigeration, and Air-Conditioning Engineers, Inc., Atlanta (1986).
10. Z□rate, L. E., Pereira, E. M., Silva, J. P., Vimieiro R., Diniz, A. S., & Pires, S. 2003a. Representation of a solar collector via artificial neural networks. In Hamza, M. H. ed. International Conference On Artificial Intelligence And Applications, Benalm□dena, Spain, 8-11 September 2003. IASTED: ACTA Press, pp. 517-522.

Appendix

Table I.1. Sample of the trainig set

T_{amb}	T_{in}	G	T_{out}
27.02	38.09	1003.38	45.04
24.99	23.72	767.35	29.95
24.07	23.06	921.81	30.35
26.76	37.92	1012.9	45
25.08	27.14	909.19	33.96
24.88	24.79	942.41	31.74
23.55	30.25	922.83	36.91
23.41	22.68	969.89	30.25
24.51	22.86	939.96	29.92
23.09	32.85	927.03	39.19

Table I.2. Comparison of efficiency values

Efficiency	ANN	Linear regression
0.649205	0.658952	0.660887
0.577049	0.565908	0.589940
0.571363	0.597309	0.581160
0.585377	0.595684	0.581360
0.619978	0.600094	0.649054
0.613144	0.657389	0.653534
0.603335	0.578951	0.607819
0.578527	0.595848	0.581105
0.657918	0.655221	0.656952
0.476594	0.561617	0.467298

Table I.3. Validation data set

T_{amb}	T_{in}	Irradiance	T_{out}	ANN T_{out}	ANN error	ANN error (%)
23.75	20.35	743.61	26.89	27.68	0.51	1.87
23.86	20.78	781.23	27.36	28.29	0.55	1.99
24.43	20.96	811.62	27.65	28.85	0.78	2.78
24.7	21.24	850.34	28.04	29.70	0.43	1.47
24.45	22.62	871.81	29.41	30.75	0.20	0.65
24.73	25.66	881.77	32.07	33.01	0.16	0.47
24.76	26.02	908.71	33.01	37.85	0.33	0.87
25.37	28.53	883.55	35.54	38.92	0.34	0.88
25.66	31.61	968.54	38.69	41.39	0.61	1.50
25.6	34.74	1007.28	41.38	41.48	0.62	1.52
25.92	34.98	1065.36	42.17	42.32	0.72	1.74
26.76	37.92	1012.9	45	42.73	0.85	2.04
26.95	38.1	1044.55	45.11	45.89	1.26	2.82
25.95	38.19	789.12	43.35	46.03	0.91	2.02
23.41	22.68	969.89	30.25	46.31	1.04	2.29
23.71	22.85	949.8	30.41	44.22	0.92	2.12
23.76	22.93	941.18	30.44	45.76	1.11	2.48
24.1	23.01	931.59	30.39	30.51	0.19	0.61
24.31	23.16	903.18	30.31	30.52	0.24	0.81
22.95	24.47	878.18	31.12	30.63	0.21	0.69
23.29	30.17	916.35	36.79	30.43	0.50	1.68
23.11	32.97	933.2	39.4	47.45	1.40	3.03
23.46	43.48	967.66	49.45	47.57	1.48	3.20
23.89	53.33	977.22	58.63	58.02	0.58	0.99
23.81	57.86	953.49	62.13	34.08	0.13	0.37
24.51	22.85	943.37	29.86	34.05	0.10	0.31
24.62	22.95	934.9	30.11	44.92	1.07	2.44
25.1	27.13	911.55	33.95	52.56	1.25	2.44
24.92	29.38	891.62	35.88	30.16	0.04	0.14
24.43	35.41	855.87	41.03	30.07	0.24	0.79

An Experiment in Task Decomposition and Ensembling for a Modular Artificial Neural Network

Brent Ferguson, Ranadhir Ghosh, and John Yearwood

School of Information Technology and Mathematical Sciences
University of Ballarat,
PO Box 663, Ballarat, Victoria 3353.
bferguson@students.ballarat.edu.au

Abstract. Modular neural networks have the possibility of overcoming common scalability and interference problems experienced by fully connected neural networks when applied to large databases. In this paper we trial an approach to constructing modular ANN's for a very large problem from CEDAR for the classification of handwritten characters. In our approach, we apply progressive task decomposition methods based upon clustering and regression techniques to find modules. We then test methods for combining the modules into ensembles and compare their structural characteristics and classification performance with that of an ANN having a fully connected topology. The results reveal improvements to classification rates as well as network topologies for this problem.

Keywords: Neural networks, modular neural networks, stepwise regression, clustering, task decomposition.

1 Introduction

Feed forward neural networks that have fully connected topologies have in the past had successful application in the problem areas of classification and regression. However their success in part favors databases where the data predominantly describes its classification as a clear function of its features. In addition it has been established in the studies of Quinlan and Collier [12], [4] that neural networks also require conditions of high feature independence to learn optimally. The use of these networks is therefore dependant upon certain conditions that the data may present itself. Adverse conditions that may degrade a neural networks performance may be resolved by appropriate structuring of the network topology.

In recent times, the structured topologies of modular artificial neural networks have attracted growing interest for overcoming the problems encountered by fully connected networks. This interest especially follows from the spiralling accumulation of complex and high dimensional data and the potential for the extraction of useful knowledge from it [5], [6], [10]. The development of modular artificial neural networks in recent reported studies suggest their appropriateness for their application in these circumstances. Modular artificial neural networks can reduce the complexity in problems arising from data having a multiple function quality that cause the condition of learning interference that occurs within fully connected topologies. The difficulty

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 97-106, 2004.

to date has been to find a means to suitably modularize network topology for large and difficult problems.

This work experiments with an approach based upon task decomposition. This is where a large task for solving a large problem is represented in terms of a number of sub tasks. Each sub task becomes a module that specialises in some part of the problem. There are two questions that arise with any approach being:

- how to decompose a problem where little or no knowledge exists?
- how many subtasks are sufficient for adequate solving of the problem?

We contribute towards answering these questions in our approach.

In this work, task decomposition is considered as the decomposition of a complex function into a number of simpler functions. Each of these smaller functions becomes the modules for our ensembles. The task decomposing process we will follow will use commonly used data processing conventions of clustering and regression and include the following operations:

- data selection
- feature selection
- feature refinement

We experiment with different methods for defining modules formed through this process and assess how they contribute overall to the performance of an ensemble of modules for several sub tasks. Where there are several modules trained for the same subtask, we trial three methods for their combination and compare the result for generalizing over test data. We then ensemble modules trained for different subtasks using a small neural network to combine their outputs and compare their characteristics with that of a fully connected neural network. The purpose of this work is to find a basis for a modularising process for large and complex tasks and an indication for the extent of task decomposition to achieve optimal classification rates. For our experiments we have chosen the problem of handwritten character classification for its suitable size and complexity.

2 Background

The early work of Rueckl and Cave [13] demonstrated the increased learning efficiency a neural network has with a modularized topology compared with one that was fully connected when learning the `what and where task'. This problem concerned itself with the recognition of a character that may present itself in one of many orientations that were represented within a grid of pixels. It was discovered that by separating the problem into two sub problems of recognition and location and by organizing the network's topology into two modules having separate hidden layers, the network benefited from improved learning and was able to classify more effectively. Following this work, there have been many investigations over the past decade into how to structure a network's topology in terms of modules to suit a variety of problems.

Schmidt [14] observes that networks of modules are either one of two types. Networks can be composed of multiple networks called modules where each has learned a separate part of the problem. These networks are commonly referred to as a mixture of experts. The outputs of each expert network are combined in a decision process that determines the contribution of each to the overall problem. The decision process may base itself upon a statistical approach such as the majority voting principle or a neural network that learns each expert's contribution such as the gating network described by Jacobs [9]. The other type of modular network described by Schmidt are those that are generally considered as not fully connected. Modules within these networks are defined as regions of the network that appear more densely connected and which are loosely connected to one another. Such networks are typical of those networks whose architectures are found though an evolutionary process. Examples of these networks appear in the work of Boers and Kuipers [2], [3], and Pal and Mitra [11]. In these networks the distribution of connections is the result of an applied genetic algorithm searching for optimal connectivity between neurons or sub network structures.

Although modular neural networks can be generally described as having a refined topology this may not necessarily mean that these networks are simplifications of the fully connected network, sometimes referred to as a monolithic neural network. Happel [8] demonstrates that for some problems, an increase in connections between neurons is needed to achieve effective learning, which results in highly complex architectures. On the other hand Boers [2] has found that modularization may not provide the most optimum topology. Amongst the population of network structures evolved for Rueckl's [13] `what' and `where' problem, a simple 2 layered artmap had produced a result similar to Rueckl's original modular solution.

In this work we organize modules into ensembles where their outputs are input to a single combining module in a similar style to the mixture of experts model. This work concentrates on the problem of modularizing the task of alphabetic character classification.

A single definition for modularity within a modular artificial neural network has not as yet been found. Modularity has mostly resulted from a task decomposition process where an overall task is divided into subtasks. A module that is implemented by a small MLP represents each subtask. An overview of the most common approaches to task decomposition and designs for modular neural networks is given by Auda and Kamel [1].

What has been consistently emerging from this area of research is their suitability for application to very large databases. Schmidt [14] has demonstrated that even by choosing the modules within a network random selection and optimization, the tendency to achieve a more efficient network grows with the size of the dataset. According to Boers [2], these networks generally have better learning efficiency with respect to training times and training stability. In general, the modular neural network achieves higher classification rates but this is dependant not only upon satisfactorily defining the modules but also according to Auda and Camel [1], on how they are to be trained and connected. Depending on one's approach on how to modularize, the result should be in terms of a highly structured topology.

Modular neural network technology may also benefit another area of research where an efficient and effective means is sought to extract an explanation from an artificial neural network trained in a particular task. A trained neural network can be regarded as a black box. How it makes a decision based on its inputs is unclear. Its knowledge is incomprehensibly represented by weight values and transfer functions. This field of research attempts to translate this sub symbolic state into interpretable rules. One of the problems here to be overcome however is also that of scalability. Large problems incur the extraction of numerous rules and the search for relevant rules becomes increasingly difficult computationally. In short, Craven [5] and Golea

[7], explain that current rule extraction techniques struggle with large fully connected networks to find compact sets of understandable rules. Pal [11] asserts that the extraction of rules can be greatly assisted by a more structured network and demonstrates the effectiveness of his modularizing technique based upon a soft computing framework of hybridized technologies.

3 Methodology

Our approach to decomposing the task of character recognition is to create modules or small neural networks that are dedicated to recognizing a particular character and that its function can distinguish this character from others. We estimate that by correctly defining the modules to preform their task then an ensemble of modules coupled with a decision network should classify characters with a greater accuracy than a fully connected neural network would. We examine the results of our experiments at two levels:

- The first level compares four cases of modularisation. The cases are sequenced to allow a series of operations to be added. These operations involve applications of clustering and regression techniques to condition data and find the inputs for modules.
- The second level begins with training a benchmark fully connected neural network trained to classify characters but primarily concerns itself with creating and training ensembles of character modules found for each case. The results for these ensembles are then compared with that obtained for the benchmark.

We analyse the results at both levels to relate the progression of module development to gains in ensemble classification accuracy. We also observe the details of topology both for character modules and for the ensembles such as the number of inputs, hid den neurons and connections they have to perform their task to assess the limits for our modularizing process.

For our experiments, we initially chose to scale the classification task of 26 alphabetic characters to 8 characters and we report the results relative to them. This subset of characters has been decided upon to contain those characters that are most represented by number of examples. Altogether there were 2462 examples to represent the 8 characters that were divided into 1462 examples for the training set and 1000 examples for the test set. The results for the experiments were averaged over 10 trials where training and test sets were drawn from randomized examples.

Module training details – Modules are implemented as three layered, feed forward and fully connected neural networks that use a sigmoid transfer function. Matlab 6.5 was used to train networks of modules in a PC environment using resilient backpropagation. The hidden layer for each neural network was grown using a succession of training cycles to add additional neurons. The number of hidden neurons was fixed when no further improvement in classification accuracy was observed.

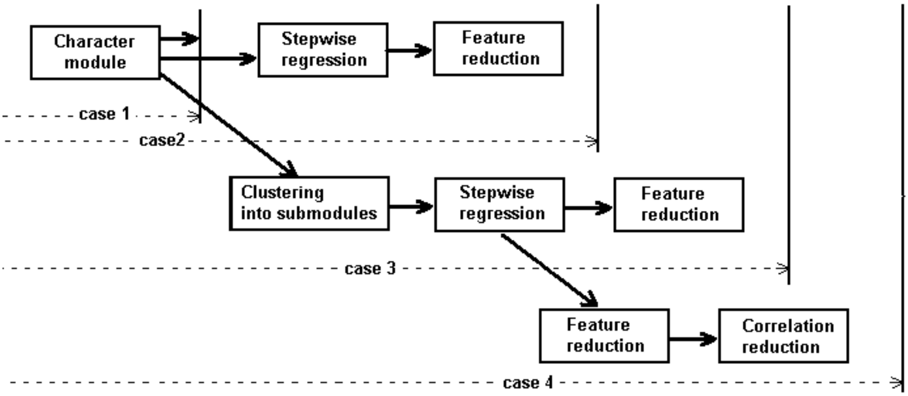


Fig. 1. Progressive module development at each case

Level 1 - Details of Module Definition Experiments

Case1 - Modularization at the character level. Modules are created for each character using all of the 100 available features of the feature vector for inputs. Each character module is trained with the training set where the examples for this character are distinguished from the others.

Case 2 - Modularization at the character level with feature reduction. In this case modules are created for each character with a reduced number of inputs. This follows with the application of a stepwise regression algorithm to find the most relevant subset of features that are associated with the examples for a particular character. The algorithm proceeds in steps in entry mode that adds features one at a time if the significance of the subset is improved by 0.05 or more and rejects the inclusion of a feature if the significance alters in excess of 0.1.

Case 3 - Modularization within character level with feature reduction. This case explores modularization within the examples for a character. Fig. 1 illustrates the process for defining case 3 modules. The examples are clustered with a self organizing map SOM into 4 groups. The value of 4 has been chosen to reflect a moderate number of clusters and is also influenced by the estimated number of handwriting styles that a character may be formed with. It is only desired to observe the usefulness of clustering at this point in developing our approach and the determination of an optimal number will be left for future consideration. The inputs for each cluster are found through stepwise regression similarly to the modules of case 2. For the purposes of regression, the examples for this cluster are distinguished from the remainder of the training and test sets, which also includes the examples from the other 3 clusters. Modules so defined become the submodules of the decision module. Each submodule undergoes the training process until complete. At this stage the training set is propagated through the submodules where the decision module trains to associate their outputs with the character class.

Case 4 - Modularization within character level with feature reduction and correlation reduction. Case 4 modules have been defined similarly as for case 3 modules with the addition of further reduction of the input feature set by removing those features that are highly correlated to another. Referring to fig. 1 for case 4 module definition an additional process followed feature reduction by stepwise regression. This consideration investigates the conditions with which correlated features can be removed. This process involves searching a correlation matrix produced for the feature subset found in the previous step for highly correlated feature pairs. Three experimental rules were constructed for deciding which feature should be removed from the subset.

Let A, B be feature pairs having a correlation in excess of 0.7.

- If A is correlated to any other feature above lower limit=0.5 then remove A from feature subset.
- If B is correlated to any other feature above lower limit=0.5 then remove B from feature subset.
- A or B is not correlated to any other feature above lower limit then remove A from feature set if P value greater than B otherwise remove B.

The upper and lower correlation limits have been set by trial and error in prior experimental determination.

Comparison study. Modules for a particular character developed in each of the four cases are compared for their classification accuracy and structural details. That is the number of hidden neurons and network connections there are for the module to perform its subtask.

Where there are several modules existing for the same subtask such as the submodules of case 3 and case 4. A means is sought to combine their outputs into one so that a comparison can be made with case 1 and case 2 modules. We trial three different methods and select the highest classifying one for use as a decision module in our comparison tests. The three methods are outlined in fig. 2.

- Combining method 1: is based upon the majority vote principle. On the basis of two or more submodule outputs that strongly indicate the classification of a particular character, the higher value is output from the combining process otherwise the lowest value is output.
- Combining method 2: the outputs are multiplied by a weighted value. This value results from the number of training set examples there are from the clustering process that defines this submodule divided by the total number of examples for the four submodules.
- Combining method 3: a small neural network inputs the submodule outputs and trains to associate the inputs with an output character classification.

Level 2 - Module Ensemble Experiments

Level 2 looked at propagating the accuracy obtained for modules at the character level to the ensemble level. Experiments were conducted to train ensembles of all eight character modules found for each case and compare the training and test set accuracy for each case. To combine the modules for each case, a neural network module inputs their outputs and trains to learn to associate their output status with one of eight character classifications. For comparison purposes a fully connected neural network was trained with the same dataset and serves as a benchmark.

4 Results and Discussion

The tasks at level 1 to evaluate modules resulting from methods described for case 1 to case 4 modularisation with reference to fig. 2, reveal overall a tendency for improved test set accuracy for each of the cases with the exception of case 4 having a slight decrease. The immediate reduction of average connection numbers referring to fig. 3 in case 2 to that of case 1 is expected as the result of feature selection due to regression. This trend is not carried through to case 3 and case 4 where both cases show large increases in connections relative to case 1 modules.

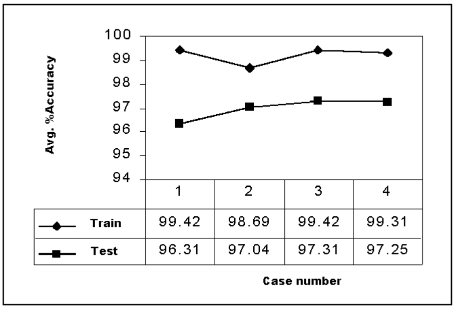


Fig. 2. Avg. module accuracy

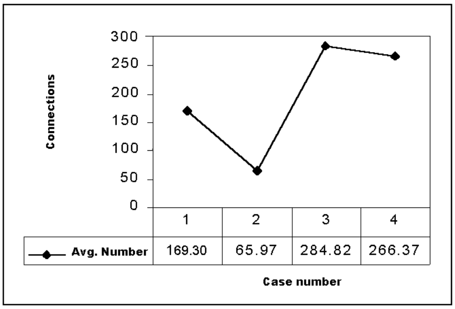


Fig. 3. Avg. module connections

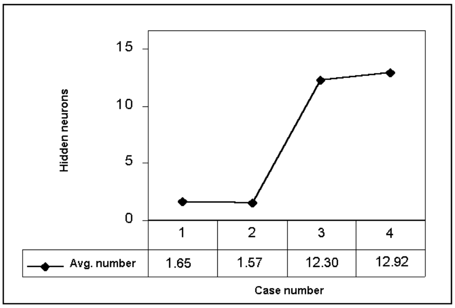


Fig. 4. Avg. hidden neurons

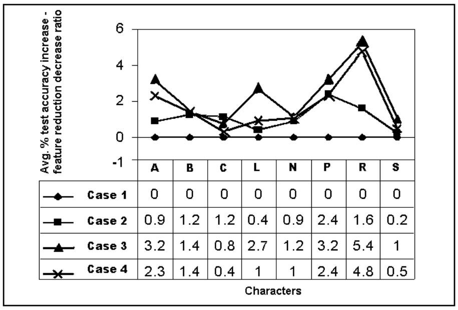
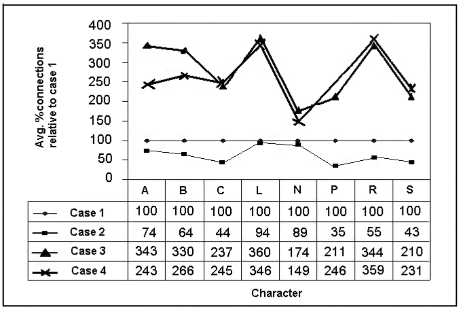
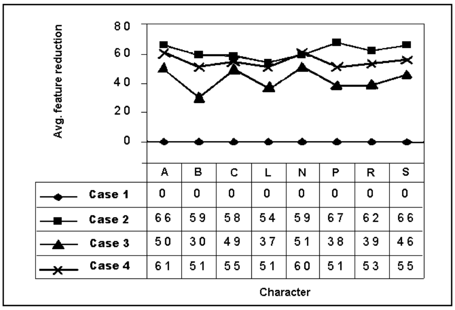


Fig. 5. Avg. reduction to features

This trend is accompanied by a sharp increase in the number of hidden neurons for both cases as indicated in fig. 4. However it was noted that the numbers of hidden neurons in a submodule of case 3 or case 4 be comparable to the number in a case 2 module. When considering case 3 submodules together, the total number of hidden neurons should therefore be optimally less than or equal to four times the number in a case 2 module. The average number of hidden neurons observed is greater than six times the number, which may suggest the presence of a false cluster. This circumstance may well improve by decreasing the number of clusters.



	A	B	C	L	N	P	R	S
◆ Case 1	100	100	100	100	100	100	100	100
■ Case 2	74	64	44	94	89	35	55	43
▲ Case 3	343	330	237	360	174	211	344	210
✕ Case 4	243	266	245	346	149	246	359	231

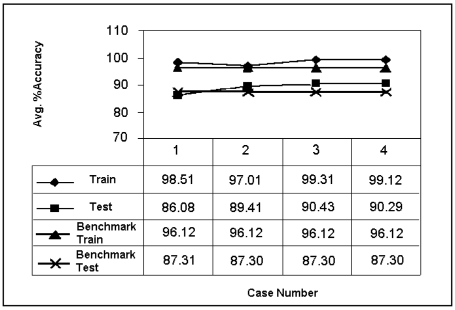


	A	B	C	L	N	P	R	S
◆ Case 1	0	0	0	0	0	0	0	0
■ Case 2	66	59	58	54	59	67	62	66
▲ Case 3	50	30	49	37	51	38	39	46
✕ Case 4	61	51	55	51	60	51	53	55

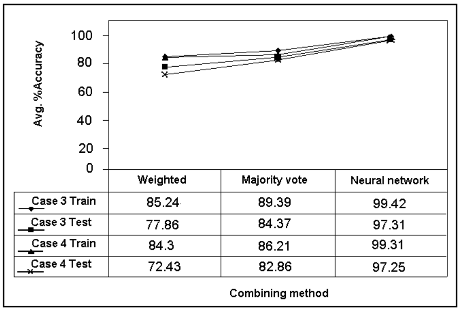
Fig. 6. Avg. improvement to connections

Fig. 7. Avg. feature reduction

When assessing the possible gain in terms of test accuracy improvement to feature reduction for modules relative to case 1 in fig. 6, case 4 modules appeared to follow the trend of case 3 modules for six of the eight characters with a slightly lower ratio. Considering this and the improved feature reduction of these modules over case 3 modules and being closer to that of case 2 modules that is indicated in fig. 7, case 4 modules may perform better given an optimal number for the submodule clustering. This would verify that further reduction of features from the input space on the basis of their correlation with other features is plausible but requires further investigation. The test classification to feature reduction ratio plotted for the four cases of character module in fig.5 indicates overall in favour of case 3 module development.



	1	2	3	4
◆ Train	98.51	97.01	99.31	99.12
■ Test	86.08	89.41	90.43	90.29
▲ Benchmark Train	96.12	96.12	96.12	96.12
✕ Benchmark Test	87.31	87.30	87.30	87.30



	Weighted	Majority vote	Neural network
◆ Case 3 Train	85.24	89.39	99.42
■ Case 3 Test	77.86	84.37	97.31
▲ Case 4 Train	84.3	86.21	99.31
✕ Case 4 Test	72.43	82.86	97.25

Fig. 8. Avg. ensemble accuracy

Fig. 9. Avg. Combination accuracy

Combining the sub modules of case 3 and case 4 immediately shows a marked loss of test set accuracy for both the summing of weighted inputs method and the majority voting process. See fig. 8. Although the voting process improves the result over the weighting method, it does not preserve the learning at the sub module level. Linear recombination of sub module outputs does not appear to be supported for this dataset. The use of a neural network to combine the submodule outputs outperformed the other two methods and was the choice to use for the comparison tests between cases.

In level 2 experiments for module ensembles, the test set results in fig. 8 indicate improving accuracy from case 1 through to case 3. This trend also improves upon the benchmark result.

5 Conclusion

In this study we have applied our approach to successfully decompose the problem of handwritten alphabetic character classification. We have decomposed the transitional representation of this dataset into components we refer to as modules. The ensembles of modules found for each defining case, show a comparative increase in test set classification accuracy favouring modules found using both clustering and step wise regression to those found using only regression. Further, the ensembles improve upon the benchmark fully connected neural network when comparing test set accuracy and topology. The case three ensemble having an average test set accuracy of 90.43% and the benchmark having 87.3%. The case four ensembles show the likely possibility for improved structure with a comparative reduction of inputs with a reasonable preservation of test accuracy. At present you task decomposition approach is limiting at case 3 modularisation.

The result of the trial that compared methods for combining submodules developed for the same task showed in favour of using a small neural network to combine the submodule inputs. Case three modules produced an indicative comparative result with an average test score of 99.42% compared with 89.39% for the majority vote and 85.24% for weighted contribution.

In summing our approach we follow a two fold process of module creation and module refinement. The approach separates the examples associated with each character into subsets. Submodules are formed initially by clustering each subset and then a step-wise regression procedure is applied to refine the module inputs. Suitably sized neural networks are then found to represent the submodules and to combine their outputs, for both submodules having a common subtask and for ensembles of modules having different subtasks

The success of our approach has been observed for the reduced dataset of handwritten characters and the implication for the construction of a complete artificial modular neural network classifier for 26 characters is supported by the results of our experiment.. It is expected that an improvement to both network topology and generalization will result over the use of one large fully connected neural network.

Further work needs to be undertaken to confirm our method to assess its suitability for broader application. Tests will need to be carried out with different representations of data for other problems of varying dimensionality. Our approach is expected to improve from further experimentation to find an optimal number of clusters for each subtask.

References

1. Auda, G., Kamel M., Modular Neural Networks: A Survey, International Journal of NeuralSystems, Vol 9, (1999) 129-151.
2. Boers, E.J.W., Kuipers, H., Biological Metaphors and the Design of Modular Artificial Neural Networks, Masters Thesis, Leiden Univesity, Netherlands, (1992).
3. Boers, E.J.W., Kuipers, H., Happel, B.L.M., Sprinkhuizen-Kuyper, I.G., Designing Modular Artificial Neural Networks, Computing Science in the Netherlands, Proc.(CSN'93),Ed.: H.A.Wijshoff, Strichting Mathematisch Centrum, Amsterdam, (1993) 87-96.

4. Collier, P.A., Waugh, S.G., Characteristics of Data Suitable For Learning With Connectionist and Symbolic Methods, Dept. of Computer Science, University of Tasmania, (1994).
5. Craven, M., Shavlik, J., Rule extraction: Where do we go to from here?, Working paper 99-1, University of Wisconsin Machine Learning Group, (1999).
6. Fayyad, U. M., Data Mining and Knowledge Discovery: Making sense out of Data, IEEE Expert Intelligent Systems and Their Applications, Vol 11, (1996) 20-26.
7. Golea, M., Tickle, A., Andrews, R., Diederich, J., The truth will come to light, IEEE Transactions on Neural Newtorks, Vol 9, (1998) 1057-1068.
8. Happel, B., Murre, J. M., Design and evolution of modular neural network architectures, Neural Networks, Vol 7, (1994) 985-1004.
9. Jacobs, R.A., Jordan, M.I., Barto, A.G., Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks, Cognitive Science, Vol 15, (1989) 219-250.
10. Nayak, R., Intelligent Data Analysis: Issues and Challenges, Proc. of the 6th. World Multi Conferences on Systematics, Cybernetics and Informatics, July 14-18, 2002, Florida USA.
11. Pal, S., Mitra, S., Rough Fuzzy MLP: Modular evolution, rule generation and evaluation, IEEE Transactions Knowledge and Data Engineering, Vol 15, (2003) 14-25.
12. Quinlan J.R., Comparing Connectionist and Symbolic Learning Methods, Dept. of Computer Science, University of Sydney, (1993).
13. Rueckl, J., Cave, K. R., Kosslyn, S. M., Why are `what' and `where' processed by separate cortical visual systems? a computational investigation, Journal of Cognitive Neuroscience, Vol 1, (1989) 171-186.
14. Schmidt, A., A modular neural network architecture with additional generalisation abilities for high dimensional input vectors, Masters Thesis, Manchester Metropolitan University, (1996).

The Effect of Deterministic and Stochastic VTG Schemes on the Application of Backpropagation to Multivariate Time Series Prediction

Taeho Jo

SITE, University of Ottawa
800 King Edward, Canada
tjo018@site.uottawa.ca
http://www.site.uottawa.ca/~tjo018

Abstract. Since 1990s, many literatures have shown that connectionist models, such as back propagation, recurrent network, and RBF (Radial Basis Function) outperform the traditional models, MA (Moving Average), AR (Auto Regressive), and ARIMA (Auto Regressive Integrated Moving Average) in time series prediction. Neural based approaches to time series prediction require the enough length of historical measurements to generate the enough number of training patterns. The more training patterns, the better the generalization of MLP is. The researches about the schemes of generating artificial training patterns and adding to the original ones have been progressed and gave me the motivation of developing VTG schemes in 1996. Virtual term is an estimated measurement, X(t+0.5) between X(t) and X(t+1), while the given measurements in the series are called actual terms. VTG (Virtual Term Generation) is the process of estimating of X(t+0.5), and VTG schemes are the techniques for the estimation of virtual terms. In this paper, the alternative VTG schemes to the VTG schemes proposed in 1996 will be proposed and applied to multivariate time series prediction. The VTG schemes proposed in 1996 are called deterministic VTG schemes, while the alternative ones are called stochastic VTG schemes in this paper.

1 Introduction

Time series prediction is the process of forecasting a future measurement by analyzing the pattern, the trends, and the relation of past measurements and the current measurement [1]. Time series prediction is studied in the several fields: data mining in computer science, industrial engineering, business management & administration and other fields. The domains of time series prediction are various from financial area to natural scientific area: stock price, stock price index, interest rate, exchanging rate of foreign currencies, the amount of precipitation, and so on. The traditional approaches to time series prediction are statistical models: AR (Auto Regressive) , MA (Moving Average), ARMA (Auto Regressive Moving Average), and Box-Jekins

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 107-117, 2004.

Model [1]. These models are mainly linear models and the trends of time series should be analyzed before applying them to time series prediction.

Literatures have shown that neural-based approaches, such as back propagation, RBF (Redial Basis Function), and recurrent network, outperform the traditional approaches (statistical models) in the performance of predicting future measurements. In the neural based approaches, back propagation is used most commonly; it has the ability of universal approximation [2]. In 1991, A.S. Weigend and D.E. Rumelhart proposed the first neural based approach, back propagation, to time series prediction [3]. N. Kohzadi presented that back propagation outperforms one of statistical models, ARIMA, in the performance of forecasting the price of cow and wheat flour, in 1996 [4]. D. Brownstone presented that back propagation is more excellent than multi linear regression in forecasting stock market movement [5]. M. Milliaris also presented that back propagation is over one of traditional models, Black-Scholes model, in the performance of forecasting of S&P 100 implied volatility [6]. Note that Black-Scholes model is most commonly applied to predict S&P 100 implied volatility in the statistical models [6]. A.U Levin proposed back propagation in the selection of the beneficial stocks [7]. In 1997, J. Ghosn and Y. Bengio predicted the profits of the stock using neural network [8]. So, These literatures show that neural-based approaches should replace the statistical ones to time series prediction.

The neural-based approaches require the enough length of historical data. Essentially, neural networks require many training patterns enough for the robust generalization [2]. Training patterns for the neural network, what is called time delay vectors, are generated from the time series in the training period by sliding window. The longer the historical length of time series in training period, the more the training patterns generated. The number of training patterns influences the generalization performance.

Actually, training patterns are not always given enough for the robust generalization. It is necessary to maintain the robust generalization performance, although training patterns are not enough. It is proposed that the generalization performance is improved by generating derived training patterns from the original ones and adding the derived training patterns to the original ones. Here, let's assume that the training patterns given originally are called natural training patterns, while the training patterns generated from them are called artificial training patterns. The use of both natural training patterns and artificial training patterns for training the neural network improves its generalization performance. In 1993, Abu-Mustafa proposed the use of hints, the artificial training patterns generated by the prior knowledge about the relations between input vector and output vector of the natural training patterns [9]. In 1995, Abu-Mustafa presented that hints contributed to reduce the prediction error in forecasting the exchange rate between USD (US Dollar) and DM (Deuch Mark) [10]. In 1994, D.A. Cohn, Z. Ghahramami, and M. J. Jordan proposed active learning, in which the neural network is trained by generating several artificial training patterns from each natural training pattern, simultaneously [11]. In 1995, A. Krogh and J. Vedelsby applied active learning to multiple neural networks [12]. In 1996, G. An proposed the scheme of generating artificial training patterns by adding noise to each natural training pattern and training the neural network with both artificial ones and natural ones [13]. And he validated that his scheme contributed to reduce the generalization error through the sine function approximation and digit recognition [13]. In 1997, S. Cho, M. Jang, and S. Chang proposed the scheme of training neural network with the natural training patterns and the artificial ones. The artificial training patterns are called virtual samples, in which input pattern is randomized and the output pattern is determined with the committee of neural networks [14]. D. Saad and S.A. Solla applied the combination of An's scheme and weight elimination in the process of training the neural network [15]. Y. Grandvalet, S. Canu, and S. Boucheron proved that G. An's scheme improve the generalization performance theoretically [16].

As mentioned above, Abu-Mustafa's scheme needs the prior knowledge about the natural training patterns; this scheme can not work without the prior knowledge. Actually, the prior knowledge to generate hints is not always given. In the D.A. Cohn's scheme and Cho's scheme, the generation of artificial training patterns is rule of sum and very heuristic. The effect of both schemes depends on the process of generating the artificial training patterns. Except Abu-Mustafa's scheme, almost mentioned schemes are validated through toy experiments: the function approximation [13][14] and robot arm kinematics [14].

In 1996, T.C. Jo proposed VTG (Virtual Term Generation) schemes of improving the precision of time series prediction by estimating the midterm X(t+0.5) between X(t) and X(t+1) [17]. Virtual term is the estimated value of X(t+0.5), between X(t) and X(t+1), while actual term is the given term in the time series [17]. VTG (Virtual Term Generation) means the process of estimating virtual terms, and VTG schemes are the techniques of estimating virtual terms. In 1997, T.C. Jo proposed the several schemes of estimating midterms and applied them to forecasting the annual number of sunspots [18]. All of the proposed VTG schemes contributed to reduce the predicted error [18]. In 1998, T.C. Jo applied the VTG schemes to multi-variable time series prediction: the prediction of monthly precipitation in west, middle, and east area of the State, Tennese of USA [19]. In 1999, T.C. Jo applied the VTG schemes to forecasting S&P 500 stock price index in financial area [20].

The VTG schemes in [17] and [18] are called deterministic VTG schemes in this paper. Deterministic VTG schemes means the method of estimating virtual terms with a particular equation. The deterministic VTG schemes proposed in [17] and [18] are mean method, 2nd LaGrange method, and 1st Taylor method. Mean method is the scheme of estimating the virtual term X(t+0.5) by averaging the adjacent actual terms, X(t) and X(t+1). Second LaGrange method is the scheme of estimating virtual terms with the equation derived from 2nd Lagrange interpolation. All of deterministic VTG schemes reduced the prediction error compared with the case of naïve neural-based approach: the neural-based approach to time series without VTG.

In this paper, alternative VTG schemes will be proposed and compared with the deterministic ones. These VTG schemes are stochastic ones: uniform VTG scheme, normal VTG scheme, and triangle VTG scheme. Stochastic VTG schemes are the methods of estimating virtual terms with random value, while deterministic VTG schemes do not use random values to estimate virtual terms. The estimated values are variable to each trial of VTG with same scheme in stochastic VTG schemes, while the estimated values are constant to each trial of VTG with same scheme in deterministic

VTG schemes. The advantage of stochastic VTG schemes over the deterministic VTG schemes is the diversity of virtual terms with same scheme. This advantage means that the stochastic VTG schemes have the potential possibility of optimizing the estimated values of virtual terms with several trials or evolutionary computation. Another advantage over the deterministic VTG schemes is simplicity in its application to VTG, except mean method. Both second LaGrange method and first order Taylor method are more complicated than mean method or the stochastic VTG schemes. The estimated value of each virtual term is between two adjacent actual terms; the value is almost mean of them. In this paper, the basis of the stochastic VTG schemes is mean method in deterministic VTG schemes; the estimated value of each virtual term is determined by adding the mean of the adjacent actual terms and random value. The stochastic VTG schemes proposed in this paper are uniform method, normal method, and triangle method, based on the distribution for generating random values.

The model of neural network applied to time series prediction in this paper is backpropagation. The model, backpropagation, is used most commonly in the models of neural network. Although there are many models of neural networks in the world, backpropagation is applied to majority of fields in supervised learning. The reason of using backpropagation commonly is that the model is implemented most easily in the models of neural network in the world. The learning algorithm of backpropagation will be included in [2], and skipped in this paper.

In the organization of this paper, both kinds of VTG schemes will be described in the next section. In third section, two application methods of backpropagation to multivariate time series prediction, separated method and combined method, will be described [21]. In fourth section, conditions, procedure, and results of experiment to compare couple kinds of VTG schemes will be presented. The data used in the experiment is the artificial time series generated from a dynamic system, Mackay Glass equation. In the fifth section, the meaning and discussion of this studies and remaining tasks to improve the proposed scheme will be mentioned as the conclusion of this paper.

2 VTG Scheme

In this section, the schemes of estimating virtual terms will be described. There are two kinds of VTG schemes; one kind is deterministic VTG schemes proposed in [17] and [18], the other kind is stochastic VTG schemes proposed in this paper. Deterministic schemes are the schemes of estimating each virtual term with a particular equation, while stochastic schemes are the schemes of estimating each virtual term with an equation and a random value. Deterministic VTG schemes consist of mean method, second order LaGrange method, and first order Taylor method. And stochastic VTG schemes consist of uniform method, normal method, and triangle method.

2.1 Deterministic VTG Scheme

This section will describe the VTG schemes proposed in 1996 and 1997: mean method, 2nd Lagrange method, and 1st Taylor method [17] [18]. In mean method, a virtual term, is estimated by averaging the two values, X(t) and X(t+0.5) like the Eq.(1).

$$\hat{X}(t+0.5)=\frac{1}{2}(X(t)+X(t+1)) \tag{1}$$

In 2nd LaGrange method, two 2nd polynomials, $P_{21}(x)$ and $P_{22}(x)$ are constructed based on Lagrange interpolation like the following this, in the assumption that the given points are $(0,\ X(t)),\ (1,\ X(t+1)),\ (2,\ X(t+2))$ and $(0,\ X(t-1)),\ (1,\ X(t)),\ (2,\ X(t+1))$.

$$\begin{aligned}P_{21}(x)=\frac{1}{2}[&(x-X(t+1))(x-X(t+2))\\&-2(x-X(t))(x-X(t+2))\\&+(x-X(t))(x-X(t+1))]\\P_{22}(x)=\frac{1}{2}[&(x-X(t))(x-X(t+1))\\&-2(x-X(t-1))(x-X(t+1))\\&+(x-X(t-1))(x-X(t))]\end{aligned}$$

The virtual term, $X(t+0.5)$ is estimated by averaging values of $P_{21}(0.5)$ and $P_{22}(1.5)$, like Eq. (2).

$$\begin{aligned}&X(t+0.5)\approx P_{21}(0.5)\approx P_{22}(1.5)\\&\hat{X}((t+0.5)=\frac{1}{2}(P_{21}(0.5)+P_{22}(1.5))---(2)\end{aligned} \tag{2}$$

But if is 0 or $T-1$, $X(t+0.5)$ is estimated with either $P_{21}(0.5)$ or $P_{22}(1.5)$.

$$\begin{aligned}&\hat{X}((t+0.5)=P_{21}(0.5) \quad \text{if } t=0\\&\hat{X}((t+0.5)=P_{21}(1.5) \quad \text{if } t=T\text{-}1\end{aligned}$$

The nodes in the output layer generated the probability of the category given the set of words as the input pattern. Therefore, output pattern is a numerical vector consisting of normalized vales from 0 to 1, like the backpropagation. The process of computing a vector consisting of probability of each category from the set of words selected from a particular document is called generalization, and will be discussed in the subsection 3.

2.2 Stochastic VTG Schemes

The base equation of stochastic VTG schemes is eq.(3).

$$\hat{X}(t+0.5)=\frac{1}{2}(X(t)+X(t+1))+\varepsilon \tag{3}$$

In the above equation, the estimated value of X(t+0.5) is the summation of a random value, ε , and the average of two adjacent actual terms. The consideration of the stochastic VTG schemes is the method of generating a random value, ε .

Uniform method of stochastic VTG schemes is the scheme of estimating each virtual term with the summation of the average of adjacent actual terms and a random value generated from the uniform distribution in figure 1.

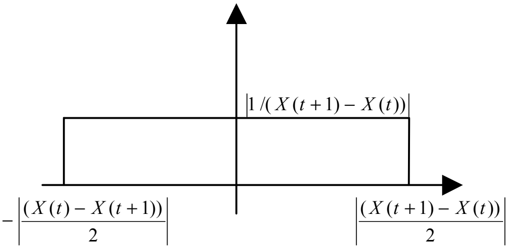

$|1/(X(t+1)-X(t))|$
$-\left|\frac{(X(t)-X(t+1))}{2}\right|$
$\left|\frac{(X(t+1)-X(t))}{2}\right|$

Fig. 1. This figure is the uniform distribution for a stochastic VTG (Virtual Term Generation) scheme

In figure 1, the estimated value of each virtual term from X(t) to X(t+1) with constant probability. The x-axis means the random value, ε, while y-axis means the probability of generating each random value ε. The probability is constant to all random values within the given range.

The second stochastic method, normal method, is the method of estimating virtual term with the summation of their average and the random value, ε, is generated based on normal distribution. In its parameters, mean is 0 and the standard deviation is $|1/2(X(t+1)-X(t))|$.

The third stochastic method, triangle method, is the scheme of estimating each virtual term with the summation of their average and the random value based on the distribution in the figure 2. Unlike the uniform distribution, this scheme has the most probability that the estimated value of a virtual term, X(t+0.5) is the average of two adjacent actual terms, X(t) and X(t+1).

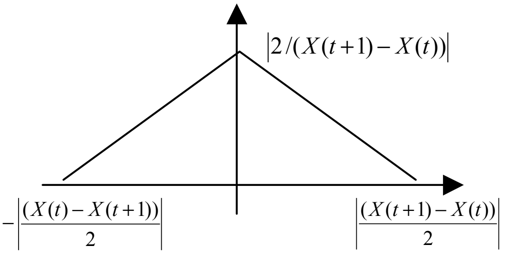

$|2/(X(t+1)-X(t))|$
$-\left|\frac{(X(t)-X(t+1))}{2}\right|$
$\left|\frac{(X(t+1)-X(t))}{2}\right|$

Fig. 2. This figure is the triangle distribution for a stochastic VTG (Virtual Term Generation) scheme

3 The Application of MLP

This phase is to predict the value of future by composing training sample from a time series including virtual terms. For first, the neural approach to time series prediction is mentioned.

Without virtual terms, the time series is like the following this.

X1(1), X1(2), ... ,X1(T)
X2(1), X2(2), ... ,X2(T)
....
Xn(1), Xn(2), ... ,Xn(T)

The training pattern from the above time series is like the following this in separate model, with the shift of 1 step[10]. These patterns are composed like that from univariate time series in [1][3]. In this case, the variables belonging to the given time series are independent among them.

input: [Xk(t-d), Xk(t-d+1), ... , Xk(t-1)]
output: Xk(t)

But in combined model, the training pattern from above time series is like the following this and the variables included in the time series are influenced among them.

input:[X1(t-d), X1(t-d+1), ... ,X1(t-1), ... , Xn(t-d), Xn(t-d+1), ... ,Xn(t-1)]
output: Xk(t)

With virtual terms, the time series is like the following this.

X1(1), X1(1.5), ... ,X1(T)
X2(1), X2(1.5), ... ,X2(T)
....
Xn(1), Xn(1.5), ... ,Xn(T)

The training pattern from the above time series including virtual terms is made like the following this in separate model. This case is same to that in univariate time series presented in [2], [12], [13], and, [19].

input: [Xk(t-d), Xk(t-d+0.5), ... , Xk(t-1)]
output: Xk(t)

In combined model, the training pattern from the time series including virtual terms is like the following this.
input: [X1(t-d), X1(t-d+0.5), ... ,X1(t-1), ... , Xn(t-d), Xn(t-d+0.5), ... ,Xn(t-1)]
output: Xk(t)

4 Experiment and Results

In order to validate the insertion of this paper, conditions, procedures, and results of this experiment will be described in this section. Both kinds of VTG schemes are applied to multivariate time series prediction, and both separated model and combined model mentioned in the precious section are applied to neural-based approaches to multivariate time series prediction. The data of time series used in this experiment is artificial time series generated from the Lorenz equation, a dynamic system. The model of neural network used in this paper is back propagation, which is used most commonly in the models of neural network. In this experiment, both VTG schemes are compared with naïve neural based approach and An's scheme of generating artificial training patterns proposed in [13].

The time series data is generated from the following equation called Lorenz equation.

$$\Delta x_1(t) = \sigma(x_2(t) - x_1(t))$$
$$\Delta x_2(t) = \rho \cdot x_1(t) - x_2(t) - x_1(t)x_3(t)$$
$$\Delta x_3(t) = x_1(t)x_2(t) - bx_3(t)$$

Three variables are determined and 1000 terms in the time series are generated from the above equation. Training period is from 1 to 700 and test period is from 701 to 1000. In the above equations, σ, ρ, and b are given parameters. In this experiment, three groups of time series data are used by modifying these parameters

The neural model of this experiment is back propagation. The architecture of this model is presented like the table 1.

Table 1. This table summarizes the architecture of back propagation in this experiment

Table 1. The architecture of back propagation in this experiment

		Input Nodes	Hidden Nodes	Output Nodes
Separated	Naïve Approach An's Scheme	10	10	1
	Both VTG Schemes	19	10	1
Combined	Naïve Approach An's Scheme	30	10	1
	Both VTG Schemes	57	10	1

In the case of time series including virtual terms, d-1 virtual terms are included in the sliding window, if the size of sliding window is d. The total number of terms within the sliding window becomes 2d-1. The learning rate of this mode is set 0.1 and the initial weight is given at random. The training epochs is fixed to 5000 in any case. The measurement of the time series prediction is prediction error, MSE (Mean Square Error)

The results of the first group of time series data are presented like the table 2. The parameters of this data is given as $\sigma = 0.99, \rho = 2.25,$ and $b = 1.775$.

Table 2. This table presents the result of the first group with parameters $\sigma = 0.99, \rho = 2.25,$ and $b = 1.775$, and each entry in this table is MSE *10^{-4}

	1st Variable		2nd Variable		3rd Variable	
	□□	□□	□□	□□	□□	□□
No VTG	4.967	1.418	4.713	1.778	2.125	1.521
An (0,0.05)	3.332	1.416	3.380	1.456	2.912	1.213
An (0,0.1)	3.313	1.319	3.295	1.434	2.918	1.092
Mean	1.085	0.9360	1.263	0.7791	1.082	0.6758
2nd Lag	1.049	0.6681	0.9181	0.9123	1.049	0.6543
1st Tayor	0.2727	0.08016	0.3318	0.4968	0.6092	0.4525
Uniform	0.9594	0.07418	1.114	0.1325	0.9645	0.8247
Gaussian	1.252	0.2341	1.198	2.919	1.055	0.9826
Triangle	0.9813	0.2312	1.462	1.578	0.9252	0.7420

The results of the second group are presented in the table 3. Its parameter of the above equation is given as $\sigma = 0.94, \rho = 2.20,$ and $b = 1.650$

Table 3. This table presents the result of the first group with parameters $\sigma = 0.94, \rho = 2.20,$ and $b = 1.650$, and each entry in this table is MSE *10^{-4}

	1st Variable		2nd Variable		3rd Variable	
	Separated	Combined	Separated	Combined	Separated	Combined
No VTG	1.842	0.9126	1.890	0.9794	1.387	2.603
An (0,0.05)	1.470	0.8828	1.459	0.9426	1.023	1.425
An (0,0.1)	1.458	0.8509	1.464	0.9321	1.012	1.490
Mean	1.074	0.6785	1.058	0.7426	0.5327	0.4990
2nd Lag	0.6791	0.4998	0.6320	0.6252	0.4644	0.4472
1st Tayor	0.7668	0.2371	0.4816	0.2746	0.4709	0.2504
Uniform	0.9206	0.5911	1.354	0.6226	0.4757	0.4980
Gaussian	0.8193	0.6206	1.383	0.6644	0.6562	0.5992
Triangle	0.9417	0.6570	0.7412	0.6804	0.6035	0.5842

The results of the third group are presented in the table 4. Its parameter of Lorenz equation is given as $\sigma = 0.90, \rho = 2.30,$ and $b = 1.0$

Table 4. This table presents the result of the first group with parameters $\sigma = 0.90, \rho = 2.30,$ and $b = 1.0$, and each entry in this table is MSE $*10^{-4}$

	1st Variable		2nd Variable		3rd Variable	
	□□	□□	□□	□□	□□	□□
No VTG	5.164	1.796	5.245	3.574	2.961	1.711
An (0,0.05)	3.739	1.796	3.845	2.834	2.028	1.469
An (0,0.1)	3.722	1.697	3.853	2.771	2.128	1.524
Mean	1.514	1.267	1.659	1.687	1.406	1.043
2nd Lag	1.436	0.8027	1.269	1.754	1.394	0.8479
1st Tayor	0.2587	0.1273	0.2016	0.2133	1.072	0.6377
Uniform	1.340	0.8361	1.348	1.579	1.434	0.9399
Gaussian	1.660	1.288	2.353	1.924	1.578	1.222
Triangle	1.883	1.294	2.089	1.821	1.550	1.066

From table2 to table 4, An's scheme improved the performance of time series prediction about 10-20% compared with naïve approach. Deterministic VTG schemes improved its performance even more than 50%; the prediction error is reduced less than half of the naïve approach and An's scheme. Stochastic VTG schemes improved its scheme compared with naïve neural-based approach and An's scheme outstandingly but are little inferior to the deterministic schemes except mean method.

5 Conclusion

This paper proposed the alternative VTG schemes and showed that these schemes reduced prediction error compared with naïve neural based approach and An's scheme and they are comparable with deterministic VTG schemes proposed [17] and [18]. The proposed VTG schemes have advantages over deterministic VTG schemes; stochastic VTG schemes have the diversity in the estimation of virtual terms. This provides the potentiality of the application of evolutionary computation to optimize virtual terms. The optimization of virtual terms means the maximization of prediction performance. Although the stochastic VTG schemes are little inferior to the deterministic VTG schemes in general, the optimization of virtual terms with evolutionary computation will improve the prediction performance compared with the deterministic VTG schemes.

References

1. Box, G.E.P., Jekins, G.M., and Reisel, G. C.: Time Series prediction. Prentice Hall, (1994)
2. Haykine, S.: Neural Networks. Macmillan Colledge Publishing Company (1994)
3. Weigend, A.S. and Rumelhart, D.E.: Generalization through Minimal Networks with Application to Forecasting. The Proceedings of 23rd Interface, (1991), 362-370

4. Kohzadi N., Boyd, M.S., Kermanshahi, B. and Kaastra I.: A comparison of artificial neural network and time series models for forecasting commodity process. Neurocomputing, Vol 10, No 2, (1996) 169-181
5. Brownstone, D.: Using percentage accuracy to measure neural network prediction on Stock Market movements. Neurocomputing, Vol 10, No 3, (1996) 237-250
6. M. Malliaris, M. and Salchenberger L.: Using neural networks to forecast the S&P 100 implied volatility. Neurocomputing, Vol 10, No 2, (1996) 183-195
7. Levin A.U.: Stock Selection via Nonlinear Multi-Layer-Factor Models. Neural Information Processing Systems, Vol 8, (1996) 966-972
8. Ghosn J. and Bengio Y.: Multi-Task Learning for Stock Selection. Neural Information Processing Systems, Vol 9, (1997) 947-952
9. Abu-Mustafa Y.S.: A Method for Learning from Hints. Neural Information Processing Systems, Vol 5, (1993) 73-80
10. Abu-Mustafa Y.S.: Financial Application of Learning from Hints. Neural Information Processing Systems, Vol 7, (1995) 411-418
11. Cohn, D.A., Ghahramami, Z., and Jordan M. J.: Active Learning with Statistical Models. Neural Information Processing Systems, Vol 6, (1994) 705-712
12. Krogh M. and Vedelsby J.: Neural Network Ensembles, Cross Validation, and Active Learning, Neural Information Processing Systems, Vol 7, (1995) 231-238
13. An, G.: The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. Neural Computation Vol 7 (1996) 643-674
14. Cho, S., Jang, M., and Chang, S.: Virtual Sample Generation using a Population of Network. Neural Processing Letters Vol 5 No 2 (1996) pp83-89
15. Saad, D. and Solla, S.A.: Learning with Noise and Regularizers in Multilayer Neural Networks. Neural Information Processing Systems Vol 9 (1997) 260-266
16. Grandvalet, Y., Canu, S., and Boucheron, S.: Noise Injection: Theoretical Prospects. Neural Computation Vol 9 (1997) 1093-1108
17. Jo, T. and Cho, S.: Time Series Prediction using Virtual Term Generation Scheme. The Proceedings of ITC-CSCC 96 (1996) pp1282-1285
18. Jo, T.: The Prediction of the Annual Number of Sunspots with Virtual Terms Generation Schemes. The Proceeding of PACES/SPICIS 97 (1997) 135-141
19. Jo, T.: The Connectionist Approach to Multi Variables Forecasting of Precipitation with Virtual Term Generation Schemes. The Proceedings of IEEE IJCNN 98 (1998) 2531-2534
20. Jo, T.: Neural Approach to Forecasting of S&P 500 Stock Price Index with Virtual Term Generation. The Proceedings of International Conference on Artificial Intelligence 99 (1999) 502-507
21. Chakraborty, K., Mehrotra, K., Mohan, C.K., and Ranka, S.: Forecasting the Behavior of Multivariate Time Series using Neural Network. Neural Network Vol 5 (1992) 961-970

Gene Discovery in Leukemia Revisited: A Computational Intelligence Perspective

Julio J. Valdés and Alan J. Barton

Institute for Information Technology, National Research Council of Canada
1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
{julio.valdes, alan.barton}@nrc-cnrc.gc.ca

Abstract. An approach using clustering in combination with Rough Sets and neural networks was investigated for the purpose of gene discovery using leukemia data. A small number of genes with high discrimination power were found, some of which were not previously reported. It was found that subtle differences between very similar genes belonging to the same cluster, as well as the number of clusters constructed, affect the discovery of relevant genes. Good results were obtained with no preprocessing applied to the data.

Keywords: computational intelligence, rough sets, clustering, virtual reality, feed-forward and probabilistic neural networks, data mining, leukemia

1 Introduction

This paper addresses the problem described in [7]: "How could an initial collection of samples from patients known to have certain types of leukemia be used to classify new, unknown samples?". Related works include [6], [5]. This paper investigates one, of the possibly many, computational intelligence approaches. Partition clustering is combined with rough sets, virtual reality data representation, generation of non-linear features and two kinds of neural networks. The goals are: to investigate the behavior of the combination of these techniques into a knowledge discovery process and to perform preliminary comparisons of the experimental results from the point of view of the discovered relevant genes.

2 Data Mining and Soft-Computing Techniques

2.1 Clustering Methods

Clustering with classical partition methods constructs crisp subpopulations (non overlapping) of objects or attributes. Two such algorithms were used in this study: the *Leader* algorithm [9], and the *convergent k-means* [1]. The leader algorithm operates with a dissimilarity or similarity measure and a preset threshold. A single pass is made through the data objects, assigning each object to the first cluster whose leader (i.e. representative) is close enough to the current object w.r.t. the specified measure

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 118-127, 2004.

and threshold. If no such matching leader is found, then the algorithm will set the current object to be a new leader; forming a new cluster. This technique is fast, however, it has several negative properties. For example, *i)* the first data object always defines a cluster and therefore, appears as a leader, *ii)* the partition formed is not invariant under a permutation of the data objects, and *iii)* the algorithm is biased, as the first clusters tend to be larger than the later ones since they get first chance at "absorbing" each object as it is allocated.

The k-means algorithm is actually a family of techniques, where a dissimilarity measure is supplied, together with an initial partition of the data (e.g. initial partition strategies include: random, the first *k* objects, *k-seed* elements, etc). The goal is to alter cluster membership so as to obtain a better partition w.r.t. the measure. Different variants very often give different partition results. However, in papers dealing with gene expression analysis, very seldom are the specificities of the k-means algorithm described. For the purposes of this study, the *convergent* k-means variant was used, which has the advantages of i) within groups sum of squares always decreases, and ii) convergence of the method if Euclidean distance is used.

2.2 Rough Sets

The Rough Set Theory [11] bears on the assumption that in order to define a set, some knowledge about the elements of the data set is needed. This is in contrast to the classical approach where a set is uniquely defined by its elements. In the Rough Set Theory, some elements may be indiscernible from the point of view of the available information and it turns out that vagueness and uncertainty are strongly related to indiscernibility. Within this theory, knowledge is understood to be the ability of characterizing all classes of the classification.

More specifically, an information system is a pair $\mathcal{A} = (U, A)$ where U is a non-empty finite set called the universe and A is a non-empty finite set of attributes such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the value set of a. For example, a decision table is any information system of the form $\mathcal{A} = (U, A \cup \{d\})$, where $d \in A$ is the decision attribute and the elements of A are the condition attributes. For any $B \subseteq A$ an equivalence relation $IND(B)$ defined as $IND(B) = \{(x, x') \in U^2 \mid \forall a \in B, a(x) = a(x')\}$, is associated.

In the Rough Set Theory a pair of precise concepts (called lower and upper approximations) replaces each vague concept; the lower approximation of a concept consists of all objects, which surely belong to the concept, whereas the upper approximation of the concept consists of all objects, which possibly belong to the concept. A *reduct* is a minimal set of attributes $B \subseteq A$ such that $IND(B) = IND(A)$ (i.e. a minimal attribute subset that preserves the partitioning of the universe). The set of all reducts of an information system $\mathcal{A}$ is denoted $RED(\mathcal{A})$. Reduction of knowledge consists of removing superfluous partitions such that the set of elementary categories in the information system is preserved, in particular, w.r.t. those categories induced by the decision attribute.

2.3 Virtual Reality Representation of Relational Structures

A *virtual reality*, visual, data mining technique extending the concept of 3D modelling to relational structures was introduced in http://www.hybridstrategies.com and [15]. It is oriented to the understanding of large heterogeneous, incomplete and imprecise data, as well as symbolic knowledge. The notion of data is not restricted to databases, but includes logical relations and other forms of both structured and non-structured knowledge. In this approach, the data objects are considered as tuples from a *heterogeneous space* [16], given by a Cartesian product of different *source* sets like: nominal, ordinal, real-valued, fuzzy-valued, image-valued, time-series-valued, graph-valued, etc. A set of relations of different arities may be defined over these objects. The construction of a VR-space requires the specification of several sets and a collection of extra mappings, which may be defined in infinitely many ways. A desideratum for the VR-space is to keep as many properties from the original space as possible, in particular, the similarity structure of the data [4]. In this sense, the role of l is to maximize some metric/non-metric structure preservation criteria [3], or minimizing some measure of information loss.

2.4 Neural Networks

Two kinds of neural networks were used in this study: a *hybrid stochastic-deterministic* feed forward network (SD-FFNN), and a *probabilistic neural network.* The SD-FFNN is a hybrid model based on a combination of simulated annealing with conjugate gradient [10], which improves the likelihood of finding good extrema while containing enough determinism. The Probabilistic Neural Network (PNN) [14] is a model based on bayesian classification using a generalization of Parzen's method for estimating joint probability density functions (pdf) from training samples. This network is composed of an input layer, a pattern layer, a summation layer, and an output layer.

3 Experimental Setup

The dataset used is that of [7], and consists of 7129 genes where patients are separated into *i)* a training set containing 38 bone marrow samples: 27 acute lymphoblastic leukemia (ALL) and 11 acute myeloid leukemia (AML), obtained from patients at the time of diagnosis, and *ii)* a testing set containing 34 samples (24 bone marrow and 10 peripheral blood samples), where 20 are ALL and 14 AML. Note that, the test set contains a much broader range of biological samples, including those from peripheral blood rather than bone marrow, from childhood AML patients, and from different reference laboratories that used different sample preparation protocols. Further, the dataset is known to have two types of ALL, namely B-cell and T-cell. For the purposes of investigation, only the AML and ALL distinction was made. The dataset distributed by [7] contains preprocessed intensity values, which were obtained by re-scaling such that overall intensities for each chip are equivalent (A linear regression model using all genes was fit to the data).

In this paper no explicit preprocessing of the data was performed, in order to not introduce bias and to be able to expose the behavior of the data processing strategy, the methods used, and their robustness. That is, no background subtraction, deletions, filtering, or averaging of samples/genes were applied.

A series of staged experiments were performed, using the training (D_{Tr}) and test (D_{Te}) data and are explained in the following subsections. Each stage feeds its results to the next stage of experiments, yielding a data analysis, processing stream. For each clustering solution, training and test subsets of the original raw data were constructed using cluster-derived leaders. The training set was discretized with a boolean reasoning algorithm, and then reducts and decision rules were computed. The test set was discretized according to the training cuts, and classified using the training decision rules (Fig-1).

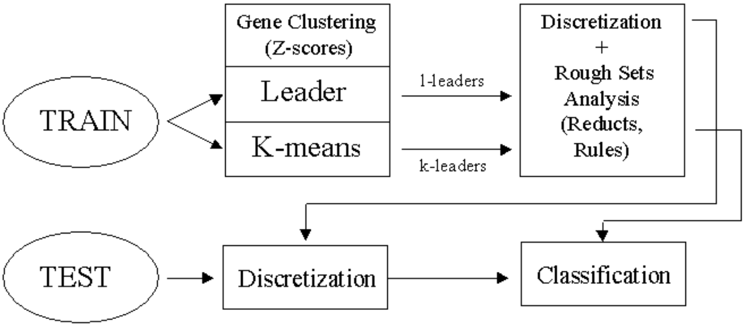

Gene Clustering (Z-scores)
Leader
K-means
TRAIN
l-leaders
k-leaders
Discretization + Rough Sets Analysis (Reducts, Rules)
TEST
Discretization
Classification

Fig. 1. Data processing strategy combining clustering with Rough Sets analysis.

Stage 1 – Selection of Representative Genes

Experimental Suite (1): [*l-leaders*]

D_{Tr} was transposed and z-score normalized to D'_{Tr}. Then the leader algorithm was applied on D'_{Tr} for the purpose of clustering the genes (using Euclidean distance and the closest leader criterion). A series of distance thresholds were used for cluster formation {0, 0.2, 0.280, 0.2805, 0.2807, 0.3, 0.4, 0.58}. Each of them induce a partition on D'_{Tr}. After that, the set of leaders were used for constructing subsets of the training data $D_{Tr,}$ referred as $D_{Tr,l\text{-}leaders}$. The same was done with the test set, D_{Te}.

Experimental Suite (2): [*k-leaders*]

For this approach D'_{Tr} was used as input to a convergent k-means algorithm with Euclidean distance, and centroid upgrading after each relocation, up to a maximum of 20 iterations (only). In order to make the results comparable with those given by the aforementioned l-leaders, the number of clusters formed (k) was chosen to be the same as those obtained for the respective l-leader. Then *k-leaders* are created from the clustering result, by selecting the closest member of the cluster w.r.t. its corresponding cluster centroid. Subsets of the training data D_{Tr} were formed, now using the k-leaders. They will be refered to as $D_{Tr,k\text{-}leaders}$ and the same k-leaders were used for constructing a subset of the test set ($D_{Te,k\text{-}leaders}$).

Stage 2 – Creation of Predictors from Training Data

Experimental Suite (1): Rough Sets methods [l-leaders]

$D_{Tr,l\text{-}leaders}$ was discretized using a boolean reasoning algorithm with a global method [13], [2] to create cut points ($C_{Tr,l\text{-}leaders}$) and a discretized form of the actual training data $D^{d}_{Tr,l\text{-}leaders}$. It is known that discretization exerts a large influence on the results of machine learning procedures, but for all experiments conducted here, the discretization method was kept constant. Rough Sets was then applied to $D^{d}_{Tr,l\text{-}leaders}$ in order to calculate the reducts; and their associated rules ($R_{Tr,l\text{-}leaders}$) were computed via an exhaustive algorithm seeking full discernibility [2]. Then, the test data D_{Te} described in terms of the same l-leaders was discretized using the cuts found for the training set ($C_{Tr,l\text{-}leaders}$) giving a discretized test set $D^{d}_{Te,l\text{-}leaders}$. Finally, the rules ($R_{Tr,l\text{-}leaders}$) found with the training data (D^{d}_{Tr}) were applied to (D^{d}_{Te}) in order to perform the classification of the new cases (computation of a confusion matrix). Global accuracies, as well the accuracies related to the individual classes are reported.

Experimental Suite (2): Rough Sets methods [k-leaders]

The same process described in Experimental Suite (1), was performed but with the k-leaders in order to obtain $C_{Tr,k\text{-}leaders}$, $D^{d}_{Tr,k\text{-}leaders}$, and the resultant $R_{Tr,k\text{-}leaders}$.

Stage 3 – Virtual Reality Representation of Data Sets

Two experiments were made:

Experiment (1): A VR-space with the union of the training (D_{Tr}) and the test (D_{Te}) sets was computed. The class membership information (ALL/AML) as well as the kind of sample (training/test) was included in the visualization. This representation is a 3-dimensional version of the original 7129-dimensional space of the raw unprocessed data. The dissimilarity measure on the original space was $((1/g)-1)$, where g is defined in [8], with a representation error defined in [12].

Experiment (2): A VR-space with the union of the sets $D_{Tr,l\text{-}leaders,}$ $D_{Te,l\text{-}leaders}$ was computed (i.e. a subset of the original gene expressions that were measured on the patient samples for both training and test). In this case, the visualization includes ALL/AML class membership, training/test data distinction information and convex hulls wrapping the AML and ALL classes allowing a better appreciation of the discrimination power of the selected genes w.r.t. the classes. The information system in the VR-space formed from the 3D-coordinates (the non-linear attributes derived from those of the original space), and the decision attribute, was used for the next processing stage.

Stage 4 – Building and Applying a Classifier to the Leukemia Data

Experiment (1): A hybrid SD-FFNN with 2 hidden nodes with hyperbolic tangent activation function, and 2 output nodes with a linear activation function was trained using mean squared error on the VR-space information system. The network was then applied to the test data set.

Experiment (2): A probabilistic neural network with 3 inputs and 2 outputs (the hidden layer comprised by each of the examples from the training set), and gaussian kernel with various variances was trained. The network was then applied to the test data set.

4 Results

The situation of the raw data (training and test sets together) as given by all of the 7129 genes is shown in Fig-2.

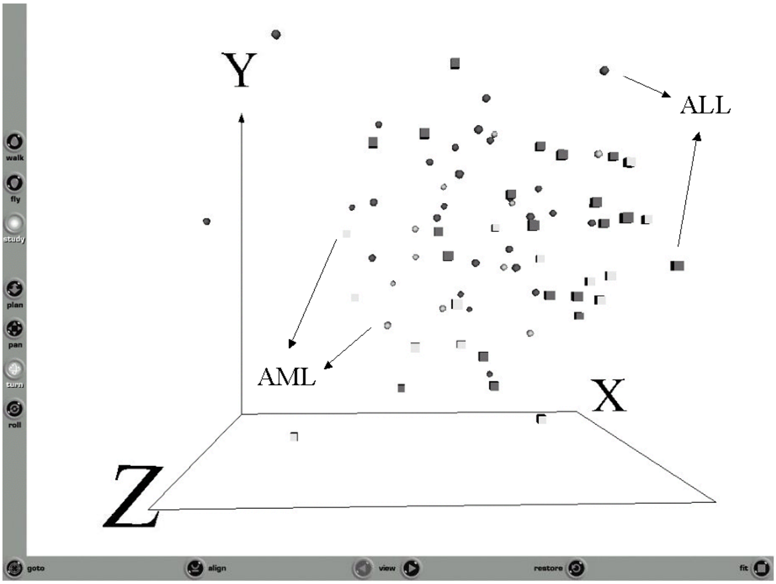


Fig. 2. Snapshot of the Virtual Reality representation of the original data (training set with 38 samples + test set with 34, both with 7129 genes). Dark objects= ALL class, Light objects=AML class. Spheres = training and Cubes = test. Representation error = 0.143, Relative error = 3.56e-6.

Despite the low representation error associated with the VR-space (which indicates that the VR representation effectively captures the overall data structure), there is no visual differentiation between the ALL and AML classes. Clearly, there are too many noisy and unrelated genes, masking and distorting the potentially relevant ones.

The results of Experimental Suite (1) (according to the tandem, Stage 1- Stage 2) are presented in Table-1. Several distance thresholds were used for partition clustering with the leader algorithm, which induced clusters of different sizes (0 distance implies using all of the original genes). It is interesting to see that despite the known limitations of the leader clustering, high accuracies are obtained with only four genes. Moreover, some of the genes are able to resolve one of the classes (ALL) perfectly, but care should be taken when interpreting these results, as criticisms questioned the correctness of the class labels of the data.

Table 1. Leader clustering results on the test set.

Distance Threshold	Nbr. of Clusters	Reducts	Accuracy		
			General	ALL	AML
0	7129	{ X95735_at}	0.912	0.9	0.929
0.2	1126	{ X95735_at}	0.912	0.9	0.929
0.280	778	{ X95735_at }	0.912	0.9	0.929
0.2805	776	{ X95735_at }	0.912	0.9	0.929
0.2807	775	{ D26308_at, M27891_at }	0.912	1	0.786
0.3	725	{ D21063_at, M27891_at }	0.853	0.95	0.714
0.4	549	{ D26308_at, M27891_at }	0.912	1	0.786
0.58	403	{ D26308_at, M27891_at }	0.912	1	0.786

When only four genes are used for describing the original data, as determined by the reducts in Table-1, the VR-space situation w.r.t. class differentiation changes completely (Fig-3). That is, a complete visual separation of the ALL and AML classes is obtained, as shown by the convex hulls wrapping the classes. Upon closer inspection, it is found that the boundary objects in the proximity zone between the two classes are test samples. Therefore indicating that resampling and cross validation could be used to improve classification errors. This is confirmed by the results of the SD-FFNN and the PNN neural network models applied to the VR-space data. Both of them had a general accuracy of 0.882 with individual accuracies of 0.9 for ALL and 0.875 for the AML classes, respectively.

The VR-space shows that the training set is more compact than the test set, confirming the biological diversity of the latter, which was previously mentioned. As described in Section 3, in this case, the attributes are the non-linear transformation of the four selected genes composed by the union of all reducts found, which minimize the similarity structure difference.

On another note, the results of Table-1 expose the dangers involved in non-careful use of clustering. That is, it is not necessarily true that similar genes imply similar relevance w.r.t. class differentiation; or in other words, just because genes are *similarly* expressed it does not mean that they may be equally useful in distinguishing between ALL and AML (maybe due to subtle differences between genes). Therefore, clustering can sometimes be misleading (see the differences between a 775 and 776 clustering solution). This effect is even more acute considering the fact that the biological literature tends to report using much smaller numbers of clusters when processing gene expression experiments.

The results of Experimental Suite (2) are presented in Table-2. The overall accuracies are higher than those obtained with the leader algorithm. Genes {X95735_at and M27891_at are found again, but two new genes emerged (X55715_at and U89922_s_at). The pair {U89922_s_at, M27891_at} was the best, from the point of view of both the general and the class-wise accuracies. The gene U14603_at is also an important complement to M27891_at, making a second best.

The situation produced by the best gene pair is depicted in Fig-2, showing that a complete class separation is potentially possible using only these two genes.

From the discovered relevant genes, {M27891_s_at} is shared with [5] and [7], {X95735_at} is shared with [7] and [6].

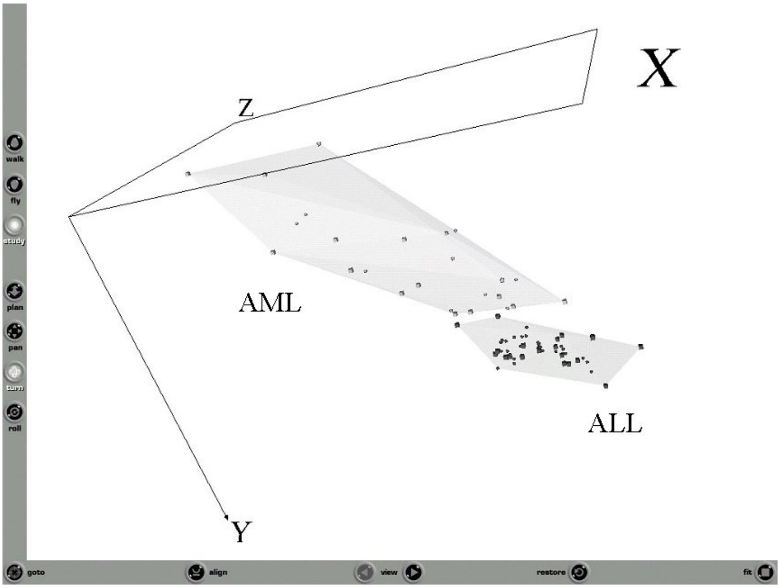


Fig. 3. Snapshot of the Virtual Reality representation of the original data (with selected genes {X95735_at, D26308_at, D21063_at, M27891_at}). Dark objects= ALL class, Light objects=AML class. Spheres = training and Cubes = test. Representation error = 0.103, Relative error = 4.63e-10.

Table 2. k-means Clustering Results on the test set

Nbr. of Clusters	Reducts	Accuracy		
		General	ALL	AML
7129	{ X95735_at}	0.912	0.9	0.929
1126	{ X95735_at}	0.912	0.9	0.929
778	{ X95735_at}	0.912	0.9	0.929
776	{ X95735_at}	0.912	0.9	0.929
775	{ X95735_at}	0.912	0.9	0.929
725	{ X55715_at, M27891_at }	0.882	0.95	0.786
549	{ U89922_s_at, M27891_at }	0.971	1	0.929
403	{ U14603_at, M27891_at }	0.941	0.95	0.929

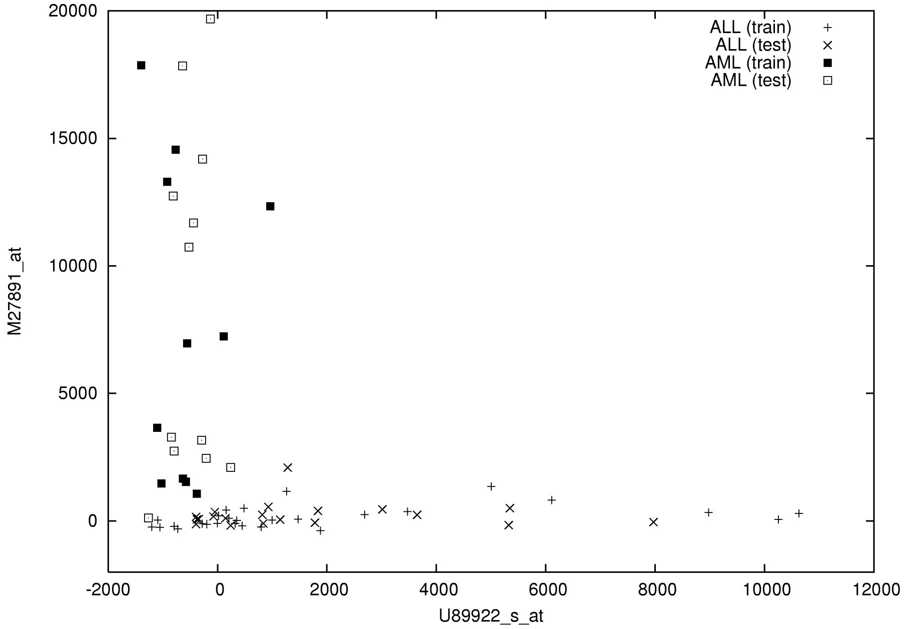


Fig. 4. Gene U89922_s_at vs. gene M27891_at for all patients in both the training and test sets. A complete separation of the ALL and AML classes is obtained.

5 Conclusions

Good results were obtained despite no preprocessing being applied to the data. Subtle differences between very similar genes belonging to the same cluster, as well as the number of clusters constructed, affect the discovery of relevant genes. Representative extraction using l or k-leaders both proved to be effective when used in tandem with Rough Sets methods; as demonstrated by the small number of genes with high discrimination power that were discovered. More thorough studies are required to correctly evaluate the impact of both the number of clusters and their generation process on the subsequent data mining steps. Also important, is the determination of appropriate ways for using these techniques in order to maximize their combined effectivity.

Visual exploration of the results (when focusing on selected genes) was very instructive for understanding the properties of the classes (size, compactness, etc.), and the relationships between the discovered genes and the classes. The visualization also helped explain the behavior of the neural network models, and suggests the potential for existence of better solutions.

Further experiments with this approach are necessary.

Acknowledgements. This research was conducted within the scope of the BioMine project (IIT-NRC). The authors would like to thank Fazel Famili, Junjun Ouyang and Robert Orchard from the Integrated Reasoning Group (IIT-NRC).

References

1. Anderberg, M.: Cluster Analysis for Applications. Academic Press, 359 pp, 1973.
2. Bazan, J.G., Szczuka S., Wroblewski, J: A New Version of Rough Set Exploration System. Third. Int. Conf. on Rough Sets and Current Trends in Computing RSCTC 2002. Malvern, PA, USA, Oct 14-17. Alpigini, Peters, Skowron, Zhong (Eds.) Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence Series) LNCS 2475, pp. 397-404. Springer-Verlag , 2002.
3. Borg, I., and Lingoes, J., Multidimensional similarity structure analysis: Springer-Verlag, New York, NY, 390 p. 1987.
4. Chandon, J.L., and Pinson, S., Analyse typologique. Théorie et applications: Masson, Paris, 254 p. 1981.
5. Deb, K and Reddy, A. R. Classification of Two-Class Cancer Data Reliably Using Evolutionary Algorithms. KanGAL Report No. 2003001. http://www.iitk.ac.in/kangal/pub.htm February, 2003.
6. Famili, F. and Ouyang, J., Data mining: understanding data and disease modeling. In Proceedings of the 21st IASTED International Conference, Applied Informatics, Innsbruck, Austria, pp. 32-37. Feb. 10-13, 2003.
7. Golub, T.R., etal. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, vol. 286, pp531-537. 1999.
8. Gower, J.C., A general coefficient of similarity and some of its properties: Biometrics, v.1, no. 27, p. 857-871. 1973.
9. Hartigan, J.: Clustering Algorithms. John Wiley & Sons, 351 pp, 1975.
10. Masters, T.: Advanced Algorithms for Neural Networks. John Wiley & Sons, pp 431,1993.
11. Pawlak, Z., Rough sets: Theoretical aspects of reasoning about data: Kluwer Academic Publishers, Dordrecht, Netherlands, 229 p. 1991.
12. Sammon, J.W. A non-linear mapping for data structure analysis. IEEE Trans. on Computers C18, p 401-409. 1969.
13. Son H. Nguyen, Hoa S. Nguyen. Discretization Methods in Data Mining. In: L. Polkowski, A. Skowron (eds.): Rough Sets in Knowledge Discovery. Physica-Verlag, Heidelberg, pp. 451-482. 1998.
14. Specht, D.: Probabilistic Neural Networks. Neural Networks 3, pp 109-118, 1990.
15. Valdés, J.J.: Virtual Reality Representation of Relational Systems and Decision Rules: An exploratory Tool for understanding Data Structure. In Theory and Application of Relational Structures as Knowledge Instruments. Meeting of the COST Action 274 (P. Hajek. Ed). Prague, November 14-16, 2002.
16. Valdés, J.J : Similarity-Based Heterogeneous Neurons in the Context of General Observational Models. Neural Network World. Vol 12, No. 5, pp 499-508, 2002.

Cell Modeling Using Agent-Based Formalisms

Ken Webb and Tony White

School of Computer Science, Carleton University, Canada
k.s.webb@sussex.ac.uk, arpwhite@scs.carleton.ca

Abstract. The systems biology community is building increasingly complex models and simulations of cells and other biological entities. In doing so the community is beginning to look at alternatives to traditional representations such as those provided by ordinary differential equations (ODE). Making use of the object-oriented (OO) paradigm, the Unified Modeling Language (UML) and Real-time Object-Oriented Modeling (ROOM) visual formalisms, we describe a simple model that includes membranes with lipid bilayers, multiple compartments including a variable number of mitochondria, substrate molecules, enzymes with reaction rules, and metabolic pathways. We demonstrate the validation of the model by comparison with Gepasi and comment on the reusability of model components.

Keywords: bioinformatics, agent-based modeling

1 Introduction

Researchers in bioinformatics and systems biology are increasingly using computer models and simulation to understand complex inter- and intra-cellular processes. The principles of object-oriented (OO) analysis, design, and implementation, as standardized in the Unified Modeling Language (UML), can be directly applied to top-down modeling and simulation of cells and other biological entities. This paper describes how an abstracted cell, consisting of membrane-bounded compartments with chemical reactions and internal organelles, can be modeled using tools such as Rational Rose RealTime (RRT), a UML-based software development tool. The resulting approach, embodied in CellAK (for Cell Assembly Kit), produces models that are similar in structure and functionality to several research tools and technologies but with greater expressive power. These include the Systems Biology Markup Language (SBML) [1], CellML [2], E-CELL [3], Gepasi [4], Jarnac [5], StochSim [6], and Virtual Cell [7]. We claim that this approach offers greater potential modeling flexibility and power because of its use of OO, UML, ROOM, and RRT. The OO paradigm, UML methodology, and RRT tool, together represent an accumulation of best practices of the software development community, a community constantly expected to build more and more complex systems, a level of complexity that is starting to approach that of systems found in biology.

All of above approaches make a fundamental distinction between structure and behavior. This paper deals mainly with the top-down structure of membranes, compartments, small molecules, and the relationships between them, but also shows

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 128-137, 2004.

how bottom-up behavior of active objects such as enzymes, transport proteins, and lipid bilayers, is incorporated into this structure to produce an executable program.

We do not use differential equations to determine the time evolution of cellular behavior, as is the case with most cell modeling systems. Differential equations find it difficult to model directed or local diffusion processes and subcellular compartmentalization [8]. Differential equation-based models are also difficult to reuse when new aspects of cell structure need to be integrated.

CellAK more closely resembles Cellulat [9] in which a collection of autonomous agents (our active objects – enzymes, transport proteins, lipid bilayers) act in parallel on elements of a set of shared data structures called blackboards (our compartments with small molecule data structures). Finally, we note that agent-based modeling of cells is becoming an area of increasing research interest and importance [8, 9].

This paper consists of 4 further sections. The next section introduces the Real-Time Object- Oriented Methodology (ROOM). A CellAK model is then described that uses the concepts of inheritance, containment, ports and connectors. Having introduced the model, a validation section is provided. The paper concludes with a review of key messages and references to future work.

2 The ROOM Formalism

David Harel, originator of the hierarchical state diagram formalism used in the Unified Modeling Language (UML) [10], and an early proponent of visual formalisms in software analysis and design [11], has argued that biological cells and multi-cellular organisms can be modeled as reactive systems using real-time software development tools [12,13].

Complexity in reactive systems arises from complicated time-varying interactions over time. The structure of a reactive system consists of many interacting components, in which control of the behavior of the system is distributed amongst the components and the locality of these interactions is of considerable importance. Very often the structure itself is dynamic, with components being created and destroyed during the system's life span, see [13 p.5].

The Rational Rose RealTime (RRT) tool is a software instantiation of Real-time Object-Oriented Modeling (ROOM) [14] methodology. Software developers design software with RRT using UML class diagrams by decomposing the system into an inheritance hierarchy of classes and a containment hierarchy of objects. Each architectural object, or capsule as they are called in RRT, contains a UML state diagram that is visually designed and programmed to react to externally generated incoming messages (generated within other capsules or sent from external systems), and to internally-generated timeouts. Messages are exchanged through ports defined for each capsule. Ports are instances of protocols, which are interfaces that define sets of related messages. All code in the system is executed within objects' state diagrams, along transitions from one state to another. An executing RRT system is therefore an organized collection of communicating finite state machines. The RRT run-time scheduler guarantees correct concurrent behavior by making sure that each transition runs all of its code to completion before any other message is processed.

The RRT design tool is visual. During design, to create the containment structure, capsules are dragged from a list of available classes into other classes. For example, the designer may drag an instance of Nucleus onto the visual representation of EukaryoticCell, thus establishing a containment relationship. Compatible ports on different capsules are graphically connected to allow the sending of messages. UML state diagrams are drawn to represent the behavior of each capsule. Other useful UML graphical tools include use case diagrams, and sequence diagrams. External C++, C, or Java classes can be readily integrated into the system.

The developer generates the executing system using visual programming, dragging and dropping objects onto a graphical editor canvas. RRT generates all required code from the diagrams, and produces an executable program. The executable can then be run and observed using the design diagrams to dynamically monitor the run-time structure and behavior of the system.

The powerful combination of the OO paradigm as embodied in the UML and ROOM visual formalisms with the added flexibility of several embedded programming languages, bundled together in a development tool such as RRT, provide much that is appropriate for biological modeling.

To summarize, benefits of the CellAK that are of use in cell and other biological modeling that have been identified so far in this paper include: support for concurrency and interaction between entities, scalability to large systems, use of inheritance and containment to structure a system, ability to implement any type of behavior that can be implemented in C, C++ or Java, object instantiation from a class, ease of using multiple instances of the same class, and subclassing to capture what entities have in common and how they differ.

3 The Model

3.1 Classes, Capsules, and Containment

The purpose of the small example system described here is to model and simulate metabolic pathways, especially the glycolytic pathway that takes place within the cytoplasm, and the TCA cycle that takes place within the mitochondrial matrix. It also includes a nucleus to allow for the modeling of genetic pathways in which changes in the extra cellular environment can effect changes in enzyme and other protein levels. The model is easily extensible, to allow for specialized types of cells.

Figure 1 shows a set of candidate entities organized into an inheritance hierarchy, drawn as a UML class diagram. Erythrocyte and NeuronCellBody are particular specializations of the more generic EukaryoticCell type. CellBilayer, MitochondrialInnerBilayer, and MitochondrialOuterBilayer are three of potentially many different subclasses of LipidBilayer. These three share certain characteristics but typically differ in the specific lipids that constitute them. The figure also shows that there are four specific Solution entities, each of which contains a mix of small molecules dissolved in the Solvent water. All entity classes are subclasses of BioEntity.

Figure 2 shows a different hierarchy, that of containment. This UML class diagram shows that at the highest level, a EukaryoticCell is contained within an ExtraCellularSolution. The EukaryoticCell in turn contains a CellMembrane, Cytoplasm, and a Nucleus. This reductionist decomposition continues for several more levels. It includes the dual membrane structure of a Mitochondrion along with its inter-membrane space and solution and its internal matrix space and solution. Part of the inheritance hierarchy is also shown in these figures. Each Membrane contains a LipidBilayer, but the specific type of bilayer (CellBilayer, MitochondrialInnerBilayer, MitochondrialOuterBilayer) depends on which type of membrane (CellMembrane, MitochondrialInnerMembrane, MitochondrialOuterMembrane) it is contained within.

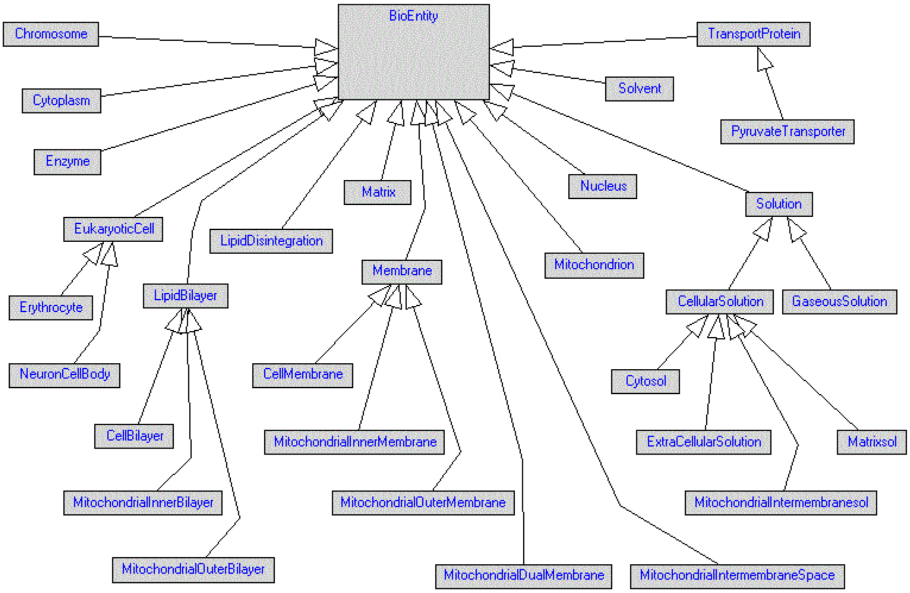


Fig. 1. UML Diagram for BioEntities

3.2 Specifying Adjacency

A model is constructed of capsules, which are instances of classes shown in Figure 1. Capsules are arranged in a containment hierarchy as shown in Figure 2. Connectivity between capsules determines adjacency; i.e. how changes in the state of one capsule affect another. Changes occur through the exchange of messages.

In a EukaryoticCell, CellMembrane is adjacent to and interacts with Cytoplasm, but is not adjacent to and therefore cannot interact directly with Nucleus. Interactions between CellMembrane and Nucleus must occur through Cytoplasm. It is important to have a structural architecture that will place those things adjacent to each other that need to be adjacent, so they can be allowed to interact.

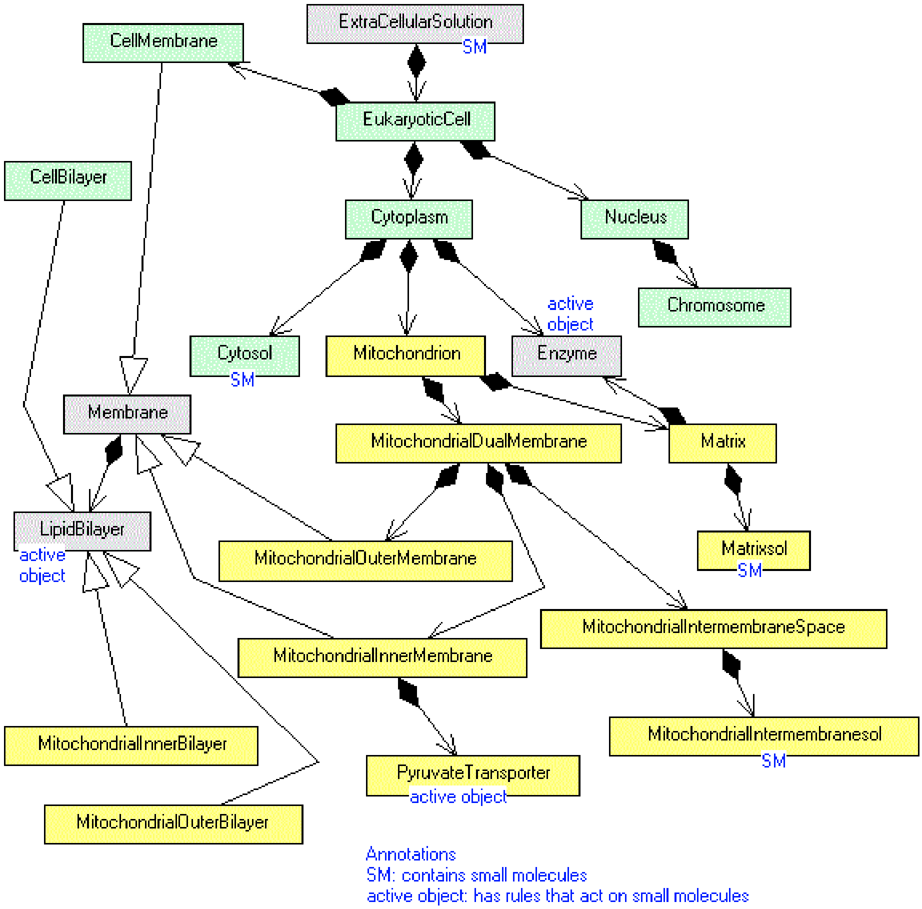

ExtraCellularSolution
SM
CellMembrane
EukaryoticCell
CellBilayer
Cytoplasm
Nucleus
Chromosome
active
object
Cytosol
SM
Mitochondrion
Enzyme
Membrane
MitochondrialDualMembrane
Matrix
LipidBilayer
active
object
MitochondrialOuterMembrane
Matrixsol
SM
MitochondrialIntermembraneSpace
MitochondrialInnerMembrane
MitochondrialIntermembranesol
SM
MitochondrialInnerBilayer
PyruvateTransporter
active object
MitochondrialOuterBilayer
Annotations
SM: contains small molecules
active object: has rules that act on small molecules

Fig. 2. Containment Hierarchy

Adjacency is represented using protocols. A protocol is a specific set of messages that can be exchanged between capsules to allow interaction. The Configuration protocol has two signals - ConfigSig and MRnaSig. When the simulation starts, the Chromosome within the Nucleus sends a ConfigSig message to the Cytoplasm, which will recursively pass this message to all of its contained capsules. When an active object such as an Enzyme receives the ConfigSig message, it determines its type and takes on the characteristics defined in the genome for that type. When a Solution such as Cytosol receives the ConfigSig message, it extracts the quantity of the various molecules that it contains, for example how many glucose and how many pyruvate molecules. In addition to being passed as messages through ports, configuration information may be also be passed in to a capsule as a parameter when it is created. This is how the entire Mitochondrion containment hierarchy is configured. In this approach, Nucleus is used for a purpose in the simulation that is similar to its actual role in a biological cell. The MRnaSig (messenger RNA signal) message can be used to reconfigure the system by creating new Enzyme types and instances as the simulation evolves in time.

The Adjacency protocol allows configured capsules to exchange messages that will establish an adjacency relationship. Capsules representing active objects (Enzymes, PyruvateTransporter and other types of TransportProtein, LipidBilayer) that engage in chemical reactions by acting on small substrate molecules, will send SubstrateRequest messages. Capsules that contain small molecules (types of Solution such as Cytosol, ExtraCellularSolution, MitochondrialIntermembranesol, Matrixsol) will respond with SubstrateLevel messages. Figure 3 is a capsule structure diagram that shows EukaryoticCell and its three contained capsules with named ports and connector lines between these ports. The color of the port (black or white) indicates the relative direction (in or out) of message movement.

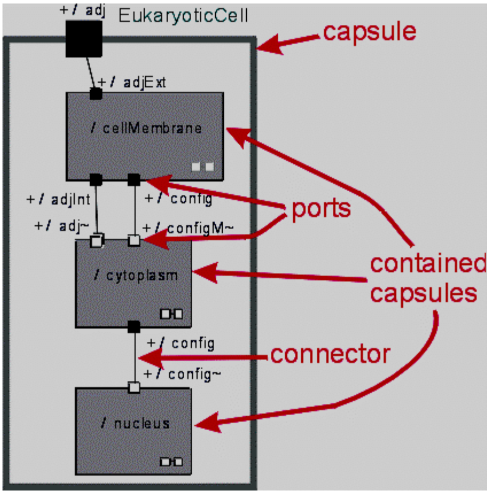


Fig. 3. Eukaryotic Cell Capsule Structure

Figure 3 represents a *significantly* simplified model; the final model includes all of the capsules shown in Figure 2. The full model was omitted here owing to space considerations.

Defining the desired behavior of the system is achieved by specifying patterns of message exchange between capsules.

In the sample model, the glycolytic pathway is implemented through the multiple enzymes within Cytoplasm, all acting concurrently on the same set of small molecules within Cytosol. The TCA metabolic pathway is similarly implemented by the concurrent actions of the multiple enzymes within Matrix acting on the small molecules of the Matrixsol. Movement of small molecules across membranes is implemented by the various lipid bilayers. For example, lipidBilayer within MitochondrialOuterMembrane transports pyruvate from the Cytosol to the MitochondrialIntermembranesol, and pyruvateTransporter within MitochondrialInnerMembrane transports pyruvate across this second membrane into the Matrixsol.

3.3 Enzyme Behaviour

Figure 4 shows the UML state diagram representing the behavior of an Enzyme active object. When first created, it makes the initialize transition. As part of this transition it executes a line of code that sends a message out its adj port. When it subsequently receives a SubstrateLevel response message through the same adj port, it stores the SmallMolecule reference that is part of that message, creates a timer so that it can be invoked at a regular interval, and makes the transition to the Active state.

The state diagrams for lipid bilayers and transport proteins are similar, but include additional states because they need to connect to two small molecule containers, one inside and the other outside.

3.4 Kinetics and Enzyme Reactions

Enzyme reactions can take various forms. In this paper, we consider the simplest case, in which an enzyme irreversibly converts a single substrate molecule into a different product molecule. More complex reactions include combining two substrates into one resulting product, splitting a single substrate into two products, and making use of activators, inhibitors, and coenzymes.

In the C++ code that implements irreversible Michaelis-Menten kinetics [15 p.148+], [4] below, `sm->` is a reference to the SmallMolecule data structure that is located in Cytosol, while `gene->` refers to a specific gene in the Chromosome. All processing by active objects makes use of these two types of data, data that they know about because of the two types of message exchange that occur during initial configuration.

```
Irreversible, 1 Substrate, 1 Product, 0 Activator, 0 Inhibitor, 0 Coenzyme
case Irr_Sb1_Pr1_Ac0_In0_Co0:
 s = sm->molecule[gene->substrateId[0]].get();
 nTimes = enzymeLevel * ((gene->substrateV * s) / (gene->substrateK + s));
 sm->molecule[gene->substrateId[0]].dec( nTimes );
 sm->molecule[gene->productId[0]].inc( nTimes );
 break;
```

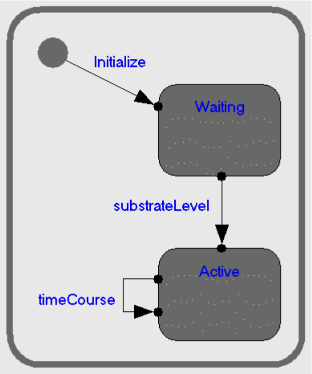


Fig. 4. Enzyme state machine

The gene in CellAK is encoded as a set of features that includes protein kinetic constants. For example, in the code above, `gene->substrateV` refers to V the upper limit of the rate of reaction, and `gene->substrateK` is the Michaelis constant K_m that gives the concentration of the substrate molecule s at which the reaction will proceed at one-half of its maximum velocity.

3.5 Validation

While the main focus in CellAK has been on a qualitative model, its accuracy is comparable to Gepasi, a tool that does claim to produce accurate quantitative results. In addition to the practical value of having CellAK generate accurate results, these also help to validate its design and implementation.

A simplified Glycolytic Pathway model was run in parallel using CellAK and Gepasi. The model includes the ten standard enzymes of glycolysis, and the eleven standard substrate and product metabolites [137 p.308]. All enzymes are implemented as irreversible, and there are no activators, inhibitors or coenzymes. Nine of the enzyme reactions convert one substrate into one product. The sole exception is the fourth enzyme reaction (Aldolase) that converts one substrate (Fructose-1,6-biphosphate) into two products (DihydroxyacetonePhosphate and Glyceraldehyde-3-phosphate). The results of this experiment are shown in Figure 5, where three of the metabolites are named and the other eight are clustered together at the bottom of the figure.

The greatest percentage difference is 4.06% for DihydroxyacetonePhosphate, the absolute quantity of which decreases to a much lower value than for any of the other metabolites. When the maximum difference is recorded at 460 seconds its value is around 2500 (2658 Gepasi, 2550 CellAK) in contrast with values between 30,000 and 230,000 for all other metabolites at that time. This relatively high difference for may be due to round-off and other errors associated with relatively small values, or it may be because of differences in the updating algorithm used at each timestep. It has been shown that there can be significant differences depending on whether synchronous or asynchronous updating is significant differences depending on whether synchronous or asynchronous updating is used [16].

There is exponentially more Glucose in the CellAK model with the passage of time than in the Gepasi version. In CellAK the cell bilayer constantly replenishes the amount of Glucose in the cytosol by transporting it at a low rate from the extra cellular solution. This low rate, as currently implemented, is not sufficient to keep the Glucose quantity constant in the cytosol. In both the Gepasi and CellAK results, the Glucose level reaches a value of around 4800 (4690 Gepasi, 4902 CellAK) after 1000 seconds.

The other metabolite with an reasonable difference is 3-PhosphoGlycerate. In both the Gepasi and CellAK results, it decreases from 100,000 to around 12,000 (12179 Gepasi, 11909 CellAK) after 1000 seconds.

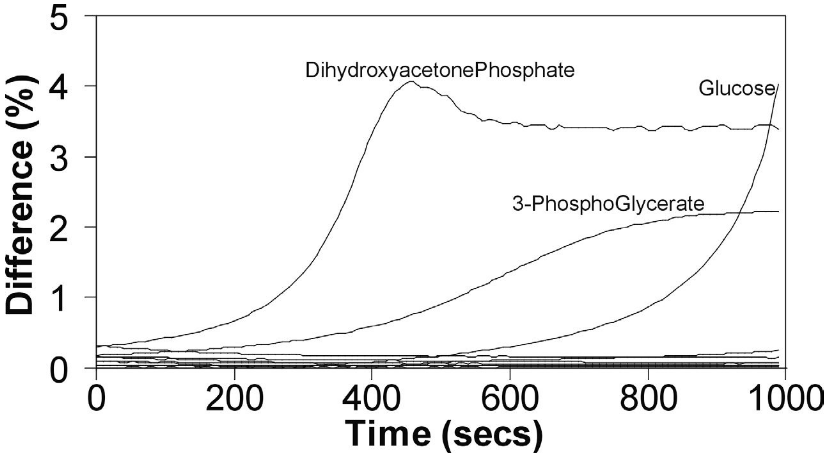


Fig. 5. CellAK vs Gepasi Results

4 Conclusions

This paper has described a modeling approach and tool, CellAK, developed using principles from agent-based modeling that is suitable for application to sophisticated cell modeling. We have demonstrated the validation of the model against Gepasi. The visual nature of the tool is considerably simpler to understand when compared to conventional differential equation based models and, being container based, can more effectively support system level models proposed by Tomita. We believe that this paper clearly confirms the value of agent-based modeling reported in [137]. Further, we have reused several of the classes and protocols in models of neurons with considerable success.

Clearly other modeling work is possible. Other active objects in CellAK (polymers) are also composed of repeating units of monomers. Becker [137 p.30] states that there are three major types of polymers in a cell. This suggests a general principle. Active objects have an influence on other active objects in CellAK by having an effect on their constituent monomers. This enhancement should now be implemented for enzymes, transport proteins, and other proteins in CellAK. However, proteins are considerably more complex than lipid bilayers. The amino acids that constitute a protein are coded for in the DNA, the order of amino acids is of critical importance, and the string of amino acids folds into a three-dimensional shape. The behavior of a protein is therefore an extremely complex function of its fine-grained structure. A more tractable problem is found in the interactions of proteins with each other, such as when one protein regulates (activates or inactivates) another protein through the process of phosphorylation [137 p.158], which involves a relatively simple reversible structural modification (a change in the fine-grained structure of another protein). The approach described in this paper could be applied relatively easily to the modeling of networks of such interacting proteins.

References

1. Hucka, M., et al., 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19, 524-531.
2. Hedley, W., et al., 2001. A short introduction to CellML. Philosophical Transactions - Mathematical Physical and Engineering Sciences 359, 1073-1089.
3. Tomita, M., et al., 1999. E-Cell: software environment for whole-cell simulation. Bioinformatics 15, 72-84.
4. Mendes, P., 1997. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. Trends. Biochem. Sci. 22, 361-363.
5. Sauro, H., 2000. JARNAC: a system for interactive metabolic analysis. Animating the Cellular Map 9th International BioThermoKinetics Meeting. University Press, ISBN 0-7972-0776-7.
6. Morton-Firth, C., Bray, D., 1998. Predicting Temporal Fluctuations in an Intracellular Signalling Pathway. Journal of Theoretical Biology 192, 117-128.
7. Loew, L., Schaff, J., 2001. The Virtual Cell: a software environment for computational cell biology. TRENDS in Biotechnology 19, 401-406.
8. Khan, S, et al., 2003. A Multi-Agent System for the Quantitative Simulation of Biological Networks. AAMAS'03, 385-392.
9. Gonzalez, P., et al., 2003. Cellulat: an agent-based intracellular signalling model. BioSystems 68, 171-185.
10. Harel, D., 1987. Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming 8, 231-274.
11. Harel, D., 1988. On Visual Formalisms. Communications of the ACM 31, 514-530.
12. Harel, D., 2003. A Grand Challenge for Computing: Full Reactive Modeling of a Multi-Cellular Animal. LNCS 2623, 2-2.
13. Kam, N., Harel, D., et al., 2003. Formal Modeling of C. elegans Development: A Scenario-Based Approach. LNCS 2602, 4-20.
14. Selic, B., Gullekson, G., Ward, P., 1994. Real-time Object-Oriented Modeling. John Wiley & Sons, New York.
15. Becker, W., Reece, J., Poenie, M., 1996. The World of the Cell, 3rd ed. Benjamin/Cummings, Menlo Park, CA.
16. Harvey, I., Bossomaier, T., 1997. Time Out of Joint: Attractors in Asynchronous Random Boolean Networks. ECAL97.

Computational Identification of RNA Motifs in Genome Sequences

Gaurav Narale[1], Jacques Beaumont[2], Philip A. Rice[1], and Mark E. Schmitt[3]

[1] Department of Chemical Engineering,
Syracuse University, Syracuse, NY 13244
{gvnarale, parice}@mailbox.syr.edu

[2] Department of Pharmacology, 750 East Adams St.
SUNY Upstate Medical University, Syracuse, NY 13210
beaumonj@upstate.edu

[3] Department of Biochemistry and Molecular Biology,
750 East Adams St., SUNY Upstate Medical University, NY.
schmittm@upstate.edu

Abstract. Many small RNAs have known 2-D structural elements. The aim of the project is to search and identify new potential small RNAs and regulatory elements in genomic databases using known 2-D conformations. The first experimental implementation was to attempt to identify the yeast mitochondrial 5S rRNA in the yeast, *Saccharomyces cerevisae*, nuclear genome. A *descriptor* was designed to identify sequences that maintain most of the conserved elements in all known 5S rRNAs. This search identified only one compatible sequence in the yeast genome. This sequence fell within a non-coding region of chromosome V. Experimental verification of this RNA is being carried out. A second experiment is to identify RNase MRP/P homologs. A descriptor was designed to search for the highly conserved cage structure. Extensive use of distributed processors was made to perform this search, dividing the sequence database into smaller chunks of individual chromosomes.

1 Introduction

RNA molecules have characteristic secondary and tertiary structures, which accounts for their diverse functional activities [1]. These structures can be grouped together as a part of collection of RNA structural motifs. The known 2-D conformations of these small RNA molecules include base-paired helices, single stranded loops, internal and external bulges and pseudoknots. RNA molecule consists of base sequence and highly complex and characterized structural elements. The non-coding RNAs such as ribosomal RNAs (rRNAs) and transfer (tRNAs) and other functional RNAs such as ribonuclease (RNase) P have constrained structural motifs. The motifs may have short or long-range base pair interactions responsible for varied functional activities of these different molecules. A complete RNA molecule may be comprised of a

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 138-143, 2004.

combination of one or more of base paired helices with unpaired single stranded nucleotide regions present in between them. Highly specific structural motifs include base pairing interactions in between helices with other structures such as E-loops, tetraloop and pseudoknots.

The aim of this study was to search and identify new potential small RNAs of interest in genomic databases using conserved 2-D conformations. The approach was to encode secondary structural patterns of RNA to describe the complexities present in an efficient way and identify sequences that have the capability of adopting the given secondary structure pattern. For this purpose, a software program known as "*RNAMotif*" was implemented to search the desired RNA [1-2] structures in genomic databases. *RNAMotif* was developed by Thomas C. Macke and David A. Case at Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA [2]. The selection of *RNAMotif* was done over other similar softwares such as *RNAMOT* and *RNABOB*. This is because RNAMotif is a further extension of both these softwares and gives more flexiblity to postscreen the initial results due to incorporation of scoring section which is absent is other softwares.

RNAMotif makes use of special coding language known as a "*descriptor*". The *descriptor* provides a blueprint for the 2-D conformations that the *RNAMotif* software identifies. Thus input to *RNAMotif* is a description of structural patterns and sequence contained in that structure which forms the part of the *descriptor*. Description file consist of four sections called *parameters*, *descriptor*, *sites* and *score*. The default or global variables that are used in the whole *descriptor* module are defined in the *parameters* section. The sites section allows users to specify relations among the elements of the *descriptor* while the *score* section ranks matches to the constraint based upon criteria defined by the user.

2 Experimental Design and Result

As a positive control and for testing purposes, a *descriptor* for tRNA was encoded and *RNAMotif* was operated over whole yeast genome. Search specifications implemented for this test run were based upon the structural motifs and conserved positions present in tRNA structure. The results obtained by test runs showed that *descriptor* designed for tRNA was able to identify almost all known tRNAs in the yeast genome database [3]. Further investigation was made to determine the reason we failed to identify the few remaining known tRNAs. All of these tRNAs were determined to contain an intron. This resulted in overall increase in the length of the tRNA sequence and hence a failure to fit into the maximum length given in the search criteria.

2.1 Screening for Potential Mitochondrial 5S rRNA Encoding Sequences

The 5S rRNA is a highly conserved and ubiquitous component in bacteria, chloroplasts, and eukaryotic cytoplasmic ribosomes. However, a 5S rRNA in mitochondrial (mt) ribosomes has only been identified in a handful of organisms [4]. So far a mt 5S has been identified only in plants, certain algae and some protist. A 5S species has not been detected in either fungal or animal mitochondria. In order to identify potential mt 5S rRNAs in fungi we designed a *descriptor* for 5S rRNA taking into consideration all structural specifications and conserved positions present, including those found in the mitochondrial variants. A general 5S rRNA structure is comprised of 5 base paired helices and 9 single stranded loops. Each helix of 5S rRNA was given a reference name, h1, h2, h3, h4 and h5, and single stranded loops were named as A, B, C, D, E, F, G, H and I. The search conditions were enforced to identify only those sequences matching exactly with known motif specifications of 5S rRNA. For this, each helical and single loop element was constrained by approximate maximum and minimum length and conserved positions wherever known [5]. The conditions applied in descriptor part of *RNAMotif* for 5S rRNA in textual form are described in Table 1.

In addition, out of the sequences which matched the search criteria, only those sequences were selected which have a ''CG'' at fourth and fifth position from the last position of the C loop. The above conditions formed the descriptor and score section of *RNAMotif* software algorithm. Please refer to http://web.syr.edu/~gvnarale/appendix.html to view the exact *RNAMotif* code implemented for this search.

Table 1. Constraints used to search for mt 5S rRNA.

Feature	Minimum Length	Maximum Length
Helix h1	9	13
Loop A	3	4
Helix h2	4	6
Loop B	2	4
Helix h3	4	6
Loop C	13	20
Loop D	3	4
Loop E	1	1
Helix h4	5	10
Loop F	1	4
Helix h5	5	10
Loop G	3	8
Loop J	0	4
Loop I	0	4

The search for potential 5S rRNA using *RNAMotif* produced only one hit in whole yeast genome database. The sequence fell in the non-coding region of chromosome V. The candidate was especially promising in this context when *RNAMotif* identified only one sequence out of entire yeast genome database and the identified sequence matched all the criteria imposed in the search. In addition, the candidate RNA contained many conserved features found in other mitochondrial 5S rRNAs that were not in the search criteria. These included a loop C of exactly 13 nucleotides that is conserved in all mt 5S rRNAs (see Fig. 1). In addition there were many specific nucleotides in this loop and the adjoining helix III that are conserved throughout the mt 5Ss. We are currently in the process of verifiying expression of the candidate RNA in yeast and examining whether it localizes to the mitochondrial compartment. In addition, we are deleting the coding region in the yeast chromose and will examine the phenotype of the mutant strain. Identification of a nuclear encoded mt 5S rRNA in yeast would be exceptionally exciting. This would necessitate import of the RNA into the mitochondria for it to function with the ribosome.

2.2 Identification of RNase MRP/P Homologs

Attempts are also being made to identify more complex structures such as the RNA components of RNase MRP/P and their homologs. RNase P is universal enzyme for generating mature 5' end of tRNAs. RNase MRP has a role in processing precursor rRNA cleaving in the internal transcribed spacer 1 between 18S and 5.8S rRNA moieties [6] and in mitochondrial DNA replication [7]. The RNase MRP/P structure is complicated by the presence of a pseudoknot. This conserved “cage” structure has been extensively analyzed, and is conserved in all known P and MRP RNAs [8-10]. Similar approach was applied to search for MRP/P homologs as was applied to 5s rRNA. The algorithm was designed in a way to identify a number of helices and single strands present in known MRP structure. The bases at specific positions, which are conserved across MRP/P homologs of different species, were also enforced in the *Score* section of program. The use of parallel processing is being used for this search. The yeast database was divided into smaller chunks of individual chromosomes and *RNAMotif* was operated with different database chunk using an LSF batch system. The results from this analysis will be discussed.

3 Future Studies

After identifying a potential 5S rRNA using *RNAMotif*, the further step is to experimentally verify the identified sequence to be the mt 5S rRNA. This will include examination of expression levels, subcellular localization, deletion of the chromosomal coding region and examination of mitochondrial ribosomal association. Successful confirmation will warrant a similar analysis of other fungi and metazoans.

The presence of other RNase MRP and P homologs has long been speculated but never demonstrated [11]. In addition, standard search paramaters for these RNAs would assist their identification in newly sequenced genomes. Because of their low degree of sequence identity and high degree of structural conservation, they are ideal candidates for this analysis. Further refinement of our search parameter should allow more efficient scans without the need for parallel processing.

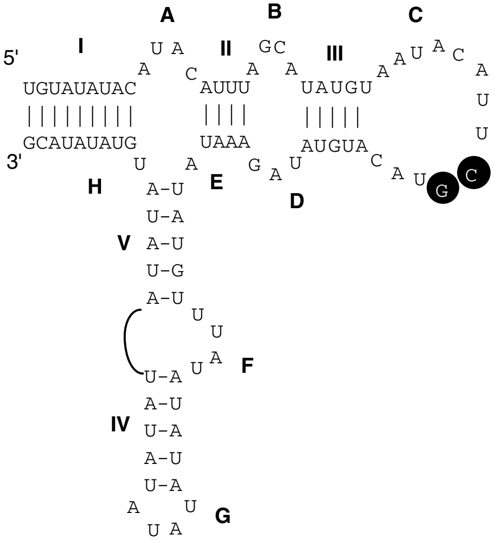


Fig. 1. Predicted structure of the potential *S. cerevisiae* mt 5S rRNA. Helixes are numbered I-V, and loops are lettered A-H. The universally conserved CG is outlined in black. Nucleotides in the conserved helix 3 and loop C that match the 5S mitochondrial consensus are in outline (Bullerwell et al., 2003).

Acknowledgements. This work was supported by grant number GM64634 from the National Institute of General Medical Sciences.

References

1. Macke, T., Ecker, D., Gutell, D. Gautheret, D. A., Case, D. A., Sampath, D. A.: RNAMotif, an RNA secondary structure definition and search algorithm. Nucl. Acids Res. 29 (2001) 4724-4735.
2 Macke, T., Case, D.: RNAMotif User's Manual, Department of Molecular Biology, The Scripps Research Institute, La Jolla, CA (2001).
3. Dolinski, K., Balakrishnan, R., Christie, K. R., Costanzo, M. C., Dwight, S. S., Engel, S. R., Fisk, D. G., Hirschman, J. E., Hong, E. L., Issel-Tarver, L., Sethuraman, A., Theesfeld, C. L., Binkley, G., Lane, C., Schroeder, M., Dong, S., Weng, S., Andrada, R., Botstein, D., and Cherry, J. M.: "Saccharomyces Genome Database" `http://www.yeastgenome.org/` (2003).
4. Bullerwell, C. E., Schnare, M. N., Gray, M.W.: Discovery and characterization of Acanthamoeba castellanii mitochhondrial 5S rRNA. RNA (2003) 9: 287-292.

5. Szymanski, M., Barciszweska, M. Z., Erdmann, V. A., Barciszweska, J.: 5S ribosomal RNA Database. Nucl. Acids Res. 30 (2002) 176-178.
6. Schmitt, M. E., Clayton, D. A.: Nuclear RNase MRP is required for correct processing of pre-5.8S rRNA in *Saccharomyces cerevisiae*. Mol. Cell. Biol. 13 (1993) 7935-7941
7. Stohl L. L., Clayton, D. A.: *Saccharomyces cerevisiae* contains an RNase MRP that cleaves at a conserved mitochondrial RNA sequence implicated in replication priming. Mol. Cell. Biol. 12 (1992) 2561-2569.
8. Forster, A. C., Altman, S.: Similar cage-shaped structure for the RNA components of all ribonuclease P and ribonuclease MRP enzymes. Cell 62 (1990) 407-409
9. Schmitt, M. E., Bennett, J. L., Dairaghi, D. J., Clayton, D. A.: Secondary structure of RNase MRP RNA as predicted by phylogenetic comparison. FASEB J. 7 (1993) 208-213.
10. Schmitt, M. E.: Molecular modeling of the three-dimensional architecture of the RNA component of yeast RNase MRP. J. Mol. Biol. 292 (1999) 827-836
11. Reilly T. H., Schmitt, M. E.: The yeast, *Saccharomyces cerevisiae*, RNase P/MRP ribonucleoprotein endoribonuclease family. Mol. Biol. Reports 23 (1996) 87-93.

An Extensible Framework for Knowledge-Based Multimedia Adaptation

Dietmar Jannach, Klaus Leopold, and Hermann Hellwagner

University Klagenfurt,
9020 Klagenfurt, Austria
{dietmar.jannach, klaus.leopold, hermann.hellwagner}@uni-klu.ac.at

Abstract. Multimedia content is becoming increasingly important in many areas not only for pure entertainment but also for commercial or educational purposes like, e.g., distance learning or online training. In parallel, the rapid evolution in the hardware sector brought up various new (mobile) end user devices like pocket PCs or mobile phones that are capable of displaying such content. Due to the different capabilities and usage environments of these devices, the basic multimedia content has to be adapted in order to fit the specific devices' capabilities and requirements, whereby such transformations typically include changes in the display size or quality adaptation. Based on the capabilities of the target device that can be expressed using recent multimedia standards like MPEG-21, these adaptation steps are typically carried out by the video server or a proxy node before the data is transferred to the client. In this paper, we present a software framework and implementation of such a multimedia server add-on that advances state-of-the-art technology in two ways. First, the framework supports the integration of various (already existing) multimedia transformation tools based on declarative interface and semantic capability descriptions in a way comparable to *Semantic Web Services* approaches. Second, by using the components' capability descriptions and the usage environment of the end user device, we employ a knowledge-based planning approach for dynamically constructing and executing the needed transformation program for a specific multi-media content request.

1 Introduction

The importance of multimedia content provision over the Internet has been steadily increasing over the last years whereby the most prominent application domains can for instance be found in the entertainment sector and in the areas of distance education and online training. Quite naturally, the field will continuously evolve as bandwidth limitations will be decreasing and new (mobile) end user devices will open new opportunities for multimedia-based applications. In a traditional multimedia content provision architecture, the content, e.g., a video, is stored on one or more servers or network nodes and streamed to the client in a certain data format and encoding. However, the variety of the new end user devices, user preferences, and other application-specific requirements like mobility show that future multimedia environments must be more flexible and adapt the content to the client's needs when transferring it over the network 3.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 144-153, 2004.

In a simple scenario, a high quality video stream is stored at the server in a given format and a request for this video is received. Assume that the request comes from a mobile client that uses a PDA (Personal Digital Assistant) and a low-bandwidth connection. Then, it is not reasonable to transfer the whole video in high quality or resolution, respectively, and leave the adaptation to the client. Transferring the large video file or "fat" stream would cause higher network traffic and, on the other hand, client-side transformation would consume a large portion of the limited processing power of the handheld device. Thus, an intelligent network node like, e.g., a video server or proxy will transform the video into an appropriate format before sending it to the client. The transformation steps invoked include, e.g., spatial adaptation of the display size, quality reduction, or greyscaling.

Current and emerging multimedia standards like MPEG-4 9 and MPEG-21 2 address the adaptation problem by providing tools to perform adaptation on multimedia streams, e.g., layered encoding, fine granular scalability, or bitstream syntax description. In this paper we describe the architecture and implementation of an intelligent multimedia server add-on that is capable of dynamically adapting a given media file to the current client's needs and capabilities. Based on declarative descriptions of the usage environment of the current session and the description of an extensible set of already existing media transformation tools, our module dynamically generates and executes "transformation plans" on the multimedia content. Due to the nature of the used algorithms and the extensibility of the description formats, the system is capable of transforming content in arbitrary ways and, on the other hand, is open for the incorporation of new tools that support, e.g., emerging multimedia standards.

The paper is organized as follows. In the next section, we give an overview of the usage scenario and describe our approach by means of an example. Next, we present implementation and algorithmic details and finally discuss how the described approach for knowledge-based program construction based on predefined modules and functions is related to the field of Semantic Web Services and Software re-use.

2 Example Problem

Figure 1 shows the general architecture of multimedia content provision over the Internet, where different clients can access content that is stored on dedicated servers in the network where also the adaptation modules are located. In the following, we will describe a simple but realistic example in order to demonstrate the functionality of our generic adaptation module. For presentation purposes, we will utilize a pseudo notation instead of the technical internal representations (in XML). First, there is a description of the content file, in our case a video, expressed by MPEG-7 media description meta data 10. The description contains information about, e.g., encoding, color, or resolution of the video. On the other hand, together with the client request, there is a description of the client's preferences on these parameters, which is contained in the *Usage Environment Description* which is also part of the forthcoming MPEG-21 *Digital Item Adaptation* standard (see Table 1).

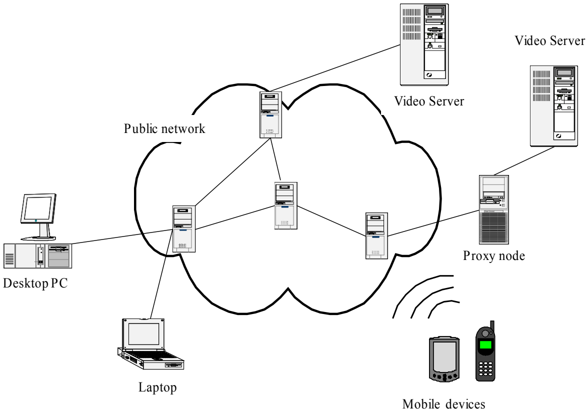


Fig. 1. Multimedia provision over the Internet

Table 1. Example for content- and usage environment descriptions

Content description:	***Usage environment description:***
type : video stream	*codec : mpeg-1*
codec : mpeg-4 Visual ES	*color : false*
color : true	*resolution : 320 x 240*
resolution : 640 x 480	

In order to transform the given video to the target state described in the *usage environment*, a set of manipulating operations have to be performed, e.g., decode the video, remove the color, change the resolution, and encode it in the required format. In fact, there are a lot of commercial and open source tools and libraries available that can perform these individual steps for different multimedia files, but in general, none of them can do all of the required transformations on different multimedia formats in one step. The main goals of our framework are to provide a mechanism to easily incorporate the functionality of different tools and to come up with a design that is general enough to cope with situations where, for instance, new description attributes or transformations are required. Therefore, we base our approach on declarative capability descriptions of the available transformation functions that are contained in libraries (plug-ins), whereby these descriptions contain the function name with the parameter descriptions as well as the preconditions and effects of the execution of the function on the content file[1]. The following example shows parts of the description of a *spatial scaler* and a *greyscaler* module.

Given all descriptions, the goal is to come up with a sequence of adaptation steps that transform the original video into the target state. In our framework we adopt a knowledge-based approach that employs a standard state-space planner (see, e.g. 4) for the computation of the plan. Once a plan is constructed, the meta-information in

[1] These descriptions of pre-conditions and effects are similar to action descriptions in classical state-space planning.

the capability descriptions is used to actually invoke the correct functions with the correct parameters from the corresponding software libraries.

module : Scaler.so
function : spatialscale
Inparameters:
fb, width, height,
newwidth, newheight
Outparameters:
newfb
Preconditions:
isFrameBuffer(fb),
codec(fb,mpeg4ves),
resolution(fb,width,height)
Effects:
isFrameBuffer(newfb),
resolution(newfb,newwidth,newheight)

module : Colorutils.so
function : greyscale
Inparameters:
fb
Outparameters:
newfb
Preconditions:
isFrameBuffer(fb),
color(fb,true)
Effects:
isFrameBuffer(newfb),
color(fb,false)

A possible plan for the example problem can be described as follows, whereby the adaptation is executed on framebuffer *fb1* and the adapted framebuffer is *fb5*: In order to fit the client's usage environments, the frame buffer is first decoded, then the size is reduced to 320x240 and color is removed, and finally the framebuffer is encoded again.

Transformation sequence:
1: decode(fb1,mpeg4ves,fb2)
2: spatialscale(fb2,640,480,320,240,fb3)
3: greyscale(fb3,fb4)
4: encode(fb4,rgb,fb5)

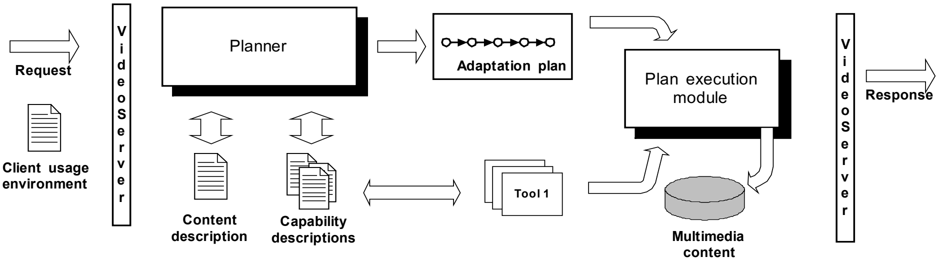


Fig. 2. Adaptation overview

Note, that besides the meta-data for generating the adaptation plan, the capability descriptions also contain a mapping from the symbolic names to concrete function implementations in the dynamically linked libraries (*.so or *.dll) and their respective parameter lists. Given this information, we are able to load the libraries and then dynamically invoke these transformation functions on the original multimedia stream. Figure 2 shows an overview of the architecture and the individual components of our framework.

3 The Planner Component

One of the major design goals for the described multimedia adaptation framework is to be as generic as possible and open for future extensions both with respect to new content adaptation tools as well as with respect to the terminology used in the capability and content descriptions. As an example, in the emerging MPEG standards, a set of terms is defined[2] that can be used to describe the client's usage environment like preferred movie genre of the user or the device's capabilities. Due to the fact that this vocabulary is subject to changes and extensions, we adopt a knowledge-based approach that is capable of operating on arbitrary sets of such symbols.

The main idea of this approach is to view the multimedia transformation task as a classical "state space planning problem" (see e.g., 4). Such a planning problem typically consists of a set of facts (ground symbols) that describe the given situation, i.e., the start state and the target situation (the *goal state*), as well as a set of possible *actions* that change the current situation. Each of the actions is described in terms of its preconditions that must be met such that the action can be applied, and a description of the effects of the execution of the action. A *plan* consists of an ordered sequence of parameterized actions that transfer a system from the start state to the goal state. The mapping of the standard planning problem to our multimedia transformation problem is quite straightforward. The start state corresponds to the description of the existing multimedia file or stream, the target state is given by the usage environment description of the current client. The actions correspond to the available multimedia transformation steps whereby the preconditions and effects are included in the tool's capability description (see examples above).

3.1 Implementation

In a first proof-of-concept implementation, we used a light-weight Prolog realization of a means-ends planner with goal regression 4[3]. Over the last years, significant advances in the field of AI-based planning were made 1 which made the state space planning approach suitable for large real-world problems. In addition, a common language for describing planning problems (PDDL - Planning Domain Description Language 7) was established in the community and is nowadays standard for most planning tools, e.g., the Blackbox 8 system. Currently, we are working on the integration of such a high-performance planner that also supports the PDDL format.

An important integration aspect for multimedia transformation planning lies in the fact that the different inputs for the planning process stem from different sources and come in different (non-PDDL) formats: Both the media description information and the usage environments are defined in an ISO-standardized XML format, i.e., they use predefined sets of tags that can be quite easily translated automatically into the required PDDL representation. At the moment, we are using a proprietary XML-based representation for the capability descriptions that is translated into the internal planner presentation. However, note that the integration of a new media

[2] The syntax is defined using XML-Schema descriptions.

[3] AMZI! Logic Server is used as the Prolog inference engine, see http://www.amzi.com.

transformation tool into our framework requires that a capability description file is made available. These descriptions have to be constructed by hand and it is the responsibility of the engineer that the preconditions and effects of the program execution are defined in a valid and complete manner. *Valid* means that only function symbols are used that are allowed in the MPEG standards, which can be checked using standard XML-validation; *complete* means that all of the required preconditions and effects are listed, whereby no automatic check is possible in general. Note however, that in the error case where the action description uses undefined symbols, the planner will produce no plans that include such an action but try to use other transformation plug-ins to reach the goal state. In fact, the planning mechanism works with arbitrary symbols such that any extensions of the allowed terms (e.g., a new codec) or plug-ins do not affect the implementation of the domain-independent planning algorithm and framework.

4 Evaluation and Optimizations

The complexity of such planning problems depends heavily on the length of the computed plans and on the number of available actions. Our experiments show that the length of the required transformation sequences is rather short, i.e., it typically does not exceed five or six steps. In our test scenario with around twenty different transformation actions, plans of such a length can be computed in a few milliseconds for all cases where a plan exists. On the other hand, the un-optimized planner implementation needs less than one second to determine that no such plan exists. Note that in general there are no hard real-time restrictions for plan generation in our application scenario because once a transformation plan is computed it is valid for a long time during video streaming. A new plan might be computed if, e.g., the client’s bandwidth changes. Nonetheless, besides the integration of a high-performance solver, other improvements in the planning component are possible. First, there are several possible plans to reach the same goal state and the *quality*, i.e., effectiveness of the plan can depend on the order of the plan steps. As an example, for performance reasons it might be reasonable to reduce the video’s size before the color is reduced, although the inverse order would also result in the same desired video format. At the moment, this knowledge is encoded as heuristics in the knowledge base but future versions will possibly require a more elaborated approach. The second research challenge concerns situations where no plan can be found that *exactly* produces the desired format. In these cases we need mechanisms to define how an *acceptable* solution for the client looks like. Other performance improvements can be reached more simply with the means of, e.g., a *plan cache*: Once a plan is computed, we can store this plan together with the client's preferences and the media content description and access the plan immediately once a similar request is received.

5 Plan Execution Module

After successful computation of the plan, the corresponding audio, video, and image processing tools have to be invoked by the *plan execution module* (see Figure 2) on the original media files in correct order using the correct parameter sets. The main aspects for developing the plan execution module are:

- We have to incorporate and utilize already existing multimedia manipulation libraries like the FFMPEG-toolkit[4].
- We need an approach where we can easily plug in new tools without changing the core that, for instance, implements new standards or faster algorithms.
- For performance and interoperability reasons, the plan execution modules have to interface to C and C++ software components.

Typically, extensibility of a framework is reached by defining a software interface that describes the function signatures that a plug-in has to implement. At run time, the registered plug-ins are loaded and only methods defined in the interface are used. The plug-in module then performs its tasks whereby the invoking software component does not have to be aware of any internals of the newly added component. Given that there are already existing libraries to be incorporated and there are lots of vendors and software providers we cannot assume that a shared interface (using C++, in our case) is available. Consequently, we would have to implement wrapper components for each of the used modules manually that map the function calls from the interface to the concrete implementations in the libraries. In order to overcome these limitations, in our framework we specify the required details like function name and parameter lists for invoking the concrete software modules in the tool's capability descriptions as described above. The major challenge in this setting lies in the fact that the plan execution module must cope with the addition of new modules and capability descriptions and the information on the low-level function invocation is only available in textual form. When using C or C++ as programming language, however, there is no built-in means for dynamic class loading as in programming languages like Java. Consequently, a wrapper component for each new tool has to be written, compiled, and linked to the plan execution module. In our framework, however, we adopt an approach that bypasses this problem by using a combination of dynamic library loading and generic argument lists construction for C functions based on a public available C library[5] where parameter lists and return value types are externally supplied in text files, i.e., in the capability descriptions of the transformation tools. The main function invocation routine for this process is sketched in Listing 1 below.

When executing the individual steps of the plan, it is important to note that the input for one function can correspond to the output of a previously called function. Typically, the already partially adapted multimedia file or stream is handed from one plan step to another. For the implementation of argument passing, we use lookup tables that contain all the symbols that are used in the generated plan that store the actual variable values after each transformation step.

[4] see http://www.ffmpeg.org

[5] See www.gnu.org,"ffcall" library

Listing 1 Dynamic function invocation

```
libname = getLibraryNameForTool(...)
funcname = getFunctionNameForTool(...)
handle = loadLibrary(libname)
f = getAddressForSymbol(handle,funcname)
for each parameter specified
    determine type and value of parameter
    add param to generic param-list
    callFunction(f,param-list)
```

6 Relation to Semantic Service Mark-Up with DAML/S

The idea of describing the capabilities of a software component, i.e., its specification, in a formal manner is not new. Besides the standard use of formal specifications in the requirements engineering phase, such specifications can for instance be used to (partially) automate the software *reuse* process (13). In these approaches, the functionality and behavior of the components of an existing software library are explicitly described using a formal language like *Z* 6 or in terms of, e.g., input-output pairs. Quite independently, in the last years another field emerged where formal descriptions of component or function behavior play an important role: *Semantic Web Services*. The goal of these efforts is to extend the *Web* which was once more or less a large repository of text and images – with *information-providing* and *world-altering* services accessible over the Internet. As these services are to be located and executed by computer programs or agents, they must be made available in a computer-interpretable way 12. While the technical basis for inter-operation with standard *Web Services* is already laid with the definition of XML-based exchange formats and protocols like SOAP and WSDL[6], a really intelligent service-providing Web requires that the properties, capabilities, and effects of execution of the available services are explicitly defined. The standard example scenario for the approach 12 is from the domain of vacation planning: the agent's goal is to compose a complete travel arrangement where the individual steps include, e.g., hotel reservations or flight booking and there are several alternative providers for hotel rooms or other required resources. Ontologies 5 – simply speaking, a common understanding of the terms and their interrelationships – are the basis for any communication in distributed and heterogeneous environments. DAML-S[7] is such an ontology having its roots in the Artificial Intelligence field and is evolving as a *de facto* standard for semantic service markup that facilitates automatic discovery, invocation, composition as well as monitoring of *Semantic Web Services* 13. DAML-S can be used to describe what a service does (*Profile*), how the service works (*Process*) and how it is actually executed (*Grounding*), whereby this information is contained in several XML resources.

Quite obviously, there is a strong relationship with our work on dynamic multimedia adaptation, particularly in the way the behavior of the services is specified

[6] Simple Object Access Protocol and Web Service Description Language (www.w3c.org)

[7] www.daml.org

(input, output, preconditions, and effects). Nonetheless, there are some important facets of the two approaches to be discussed in more detail.

*Service Discovery***.** In our approach, the complete adaptation of the multimedia content takes place on one dedicated multimedia server. Although it would in principle be possible to design simple "transformation services" as modular, distributed Web Services, we feel that such a scenario is not realistic or desirable for our application domain.

Domain ontologies. In current research efforts on Semantic Web Services the problem of establishing a common "domain ontology" is not adequately addressed. If we consider the travel arrangement planning service, it is assumed that all providers of some Web Service in the domain not only use the same vocabulary (for instance, the word/concept *reservation*) but also associate the same meaning to these terms. In the domain of video adaptation, however, we have the advantage that the terminology as well as the semantics of the involved concepts (like *resolution*) is already clearly defined in the evolving and existing MPEG multimedia standards.

Grounding. In the *grounding* part of a Semantic Web Service description, each *process* of the Web Service is related to one or more *WSDL operations* whereby the needed WSDL definitions for the low-level message exchange can be constructed from the DAML-based service descriptions. In our framework, *grounding* corresponds to the mapping of each plan step to a function call to the transformation libraries in the C/C++ libraries and the generation of adequate wrappers for module invocation.

As a result, we argue that the basic approach used for describing Semantic Web Services can be generalized and adopted in other areas of software development like software re-use based on behavior specifications or automated program construction. The main advantages of such an approach lie in the fact that with DAML and OIL (Ontology Inference Layer) an underlying mechanism and corresponding tool support for the definition and integration of domain ontologies are already available. While in our domain the application of DAML-S as core representation mechanism for describing component capabilities is quite straightforward and part of our current engineering work, our current research focuses on the generalization of the Semantic Web-Services techniques for software re-use and automated, knowledge-based program construction. We therefore see our work as a step towards this direction and the domain of multimedia adaptation as a promising field for evaluation of such an approach.

7 Conclusions

In this paper we have presented a framework that supports dynamic composition and execution of multimedia transformations using existing software libraries. Based on declarative descriptions of the available libraries and the requirements of the clients, a knowledge-based planner computes a sequence of required transformation steps. These transformations are then executed on the given multimedia source and the transformed content is then shipped to the client in the required format. The described approach was validated in a proof-of-concept implementation using standard multimedia manipulation libraries and a state-space planning engine.

References

1. Blum and M. Furst. Fast planning through planning graph analysis. *Artificial Intelligenc*e, 90:281–300, 1997.
2. J. Bormans and K. Hill, editors. *MPEG-21 Overview, ISO/IEC JTC1/SC29/WG11 N523*1. October 2002. http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm.
3. L. Böszörmenyi, H. Hellwagner, H. Kosch, M. Libsie, and S. Podlipnig. Metadata Driven Adaptation in the ADMITS Project. *EURASIP Signal Processing: Image Communication, Special Issue on Multimedia Adaptatio*n, 2003.
4. Bratko. *Prolog Programming for Artificial Intelligenc*e. Addison-Wesley, 3rd edition, 2000.
5. Chandrasekaran, J. Josephson, and R. Benjamins. What Are Ontologies, and Why do we Need Them? *IEEE Intelligent System*s, 14(1):20–26, 1999.
6. Diller. *Z: An Introduction To Formal Method*s. O'Reilly, 1996.
7. M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL, the planning domain definition language. Technical report, Yale Center for Computational Vision and Control, 1998.
8. H. A. Kautz and B. Selman. Unifying SAT-based and graph-based planning. In *Intl. Joint Conference on Artificial Intelligenc*e, pages 318–325, Stockholm, 1999.
9. R. Koenen, editor. *Overview of the MPEG-4 Standard, ISO/IEC JTC1/SC29/WG11 N466*8. March 2002. http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm.
10. J. M. Martinez. *MPEG-7 Overview, ISO/IEC JTC1/SC29/WG11 N498*0. July 2002. http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm.
11. F. Pereira and T. Ebrahimi, editors. *The MPEG-4 Boo*k. Prentice Hall PTR, 2002.
12. S. McIlraith, T.C. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems, Special Issue on the Semantic We*b, 16(2):46–53, 2001.
13. M. Sabou, D. Richards, and S. van Splunter. An experience report on using DAML-S. In *Proceedings of the World Wide Web Conference 2003, Workshop on E-Services and the Se-mantic We*b, May 2003.
14. P. Schojer, L. Böszörmenyi, H. Hellwagner, B. Penz, and S. Podlipnig. Architecture of a Quality Based Intelligent Proxy (QBIX) for MPEG-4 Videos. In *Proceedings of the World Wide Web Conference 200*3, May 2003.
15. M. Zaremski and J. Wing. Specification matching of software components. *ACM Transactions on Software Engineering and Methodolog*y, 6(4):333–369, 1997.

Methods for Reducing the Number of Representatives in Representation Choice Tasks

N.T. Nguyen and C. Danilowicz

Department of Information Systems, Wroclaw University of Technology, Poland
{thanh, danilowicz}@pwr.wroc.pl

Abstract. A representation choice problem is based on the selection of an object (or objects) called a representative (or representatives), which should at best represent a given set of objects. For solving this problem in the most of cases one defines a distance function between objects and representation choice functions (choice criteria). However, it often happens that the number of representatives is too large, what in many practical situations is not convenient for decision making processes and information management. In this paper the authors propose 3 methods for solving the problem of representatives number reduction. Some examples illustrating these methods are also included.

1 Introduction

The representation choice problem is one of the well known problems of consensus theory [1],[2]. A set *Y* can be called a representation of a set of objects (alternatives) *X*, if it consists of one or more objects (alternatives) not necessarily from *X* and, according to a particular set of previously accepted requirements, all objects from *Y* can be interpreted as the best representatives of objects from *X*. Thus an element of *Y* is called a representative of *X*.

Representation choice tasks are solved in many practical situations. To make the choice very often one can only use his (her) knowledge and intuition. For example, in the mayor election a voter votes for a candidate who is, in the opinion of the voter, a good organizer and has many experiences in managing; during a vacation stay in some place we choose those postcards which in our opinion, best describe the place for sending to our friends. However, very often without using complex methods and algorithms realization of representation choice tasks is not possible. Below we give an example.

Consider a distributed information system. In each site of this system there is an information agent, the task of which is based on selection of WWW pages and making their access for users. For realizing the choice the agent often uses the criterion of maximal relevance, which is a complex one. More concretely, for an user the agent must choose those WWW pages which in its opinion, the user will regard as relevant. Thus a chosen page is a representative of the set of all pages in the agent's database. If the sites of this system have users with identical profiles, then it is reasonable to con-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 154-163, 2004.

sider retrieval results of all agents. If in opinion of an agent the pages chosen by other agents may be relevant for some its user, then the agent should supply to the user all the pages found by its colleagues in the form of the representation of these pages. If the representation contains very large number of pages, then it is not accepted by the user because he usually expects concise information. More reservations of the user may have place when the representation elements (i.e. representatives) are not consistent thematically, because such situation means that relevant pages are dropped among non-relevant pages. This non-satisfying representation may be improved by changing choice criterion or making an additional choice by other criterion. The aim of this process is to reduce the number of representatives. One of proposed criterion may rely on such requirement that the representatives must fulfil the criterion of consistency of relevance of all agents. However, it is not good idea because experienced users know that the join of results generated by different search engines in WWW often is empty. Other solution of this problem may be based on selection of the representation on the basis of relevance confirmed by users referring to the pages supplied to them by agents. In this case it is more probably that users of other sites are satisfied with the information they obtain. Here we will not develop the subject of information search in Internet. This example is only an illustration of problems related to adjusting representatives number to realization of the aim which we want to achieve using representation choice methods. Adjusting in this sense means the reduction of representatives number.

The methods for representation reduction will be presented in Section 4. The above example also shows that representation choice methods may be useful in solving complex problems related to information retrieval in web-based systems [4]. Representation choice methods have also many other applications in practice. Here we mention some of them:

- In analysis of experts' information, where several independent experts are asked to solve the same problem. If they produce different solutions, then it is needed to determine the representation of these solutions, and next to accept one of its elements as the final solution of this problem [7];
- In weather forecasting, where weather predictions for particular regions can be constructed from sets of parameters sent by different sites of a geographical monitoring system. If some of the values of involved parameters are independently collected by different sites of this system, it can be necessary at the practical level to construct representations for existing value collections. For example, this happens when periods of rainfall for a particular geographical region are determined on the base of data coming from different sites of a water monitoring system [16],[20];
- In reconciling states of autonomous agents' knowledge of their external world. For instance, in such a system each member can be provided with a different model of current states of particular parts of the external environment. In consequence, some of these agents can believe that particular objects possess particular features, and at the same this fact can be denied by the remaining parties of this multiagent system. Therefore, in order to reach a common goal all agents can be forced to negotiate the common state of their knowledge. This means that a cer-

tain policy for determining a unified representation of a set of independent models may be needed in practical settings [6], [9].

This paper presents a brief outline of representation choice methods (Section 2). The main part is included in Sections 3 and 4, where the authors describe the representation reduction problem and present 3 methods for reduction. Some conclusions are given in Section 5.

2 Outline of Representation Choice Problem

Formally, let U denote a finite universe of objects (alternatives), and let $\Pi(U)$ denote the collection of all non-empty subsets of U. By $\hat{\prod}_k(U)$ we denote the set of k-element subsets (with repetitions) of the set U for $k\in N$, and let $\hat{\prod}(U)=\bigcup_{k>0}\hat{\prod}_k(U)$.

Each element of set $\hat{\prod}(U)$ is called a *profile* which in fact is a set with repetitions. We present here some elements of the algebra of this kind of sets. An expression $A=(x,x,y,y,y,z)$ is called a set with repetitions with cardinality equal to 6. In this set element x appears 2 times, y 3 times and z - 1 time. Set A can also be written as $A=(2*x,3*y,1*z)$. The sum of sets with repetitions is denoted by symbol $\dot{\cup}$ and is defined in the following way: if element x appears in set A n times and in B n' times, then in their sum $A\dot{\cup}B$ the same element should appear $n+n'$ times. For example, if $A=(2*x,3*y,1*z)$ and $B=(4*x,2*y)$, then $A\dot{\cup}B=(6*x,5*y,1*z)$. A set A with repetitions is a subset of a set B with repetitions ($A\subseteq B$) if each element from A does not have a greater number of occurrences than it has got in set B. For example $(2*x,3*y,1*z)\subseteq(2*x,4*y,1*z)$.

In this paper we assume that the structure of the universe U is known as a distance function δ: $U\times U\rightarrow\Re^+$, which satisfies the following conditions:

a) *Nonnegative*: $(\forall x,y\in U)[\delta(x,y)\geq 0]$

b) *Reflexive*: $(\forall x,y\in U)[\delta(x,y)=0 \text{ iff } x=y]$

c) *Symmetrical*: $(\forall x,y\in U)[\delta(x,y)=\delta(y,x)]$.

Let us notice that the above conditions are only a part of metric conditions. Metrics is a good measure of distance, but its conditions are too strong [8]. A space (U,δ) defined in this way does not need to be a metric space. Therefore we call it a *distance space*. For the use in next sections we define the following parameters:

Let $X, X_1, X_2\in\hat{\prod}(U)$, $x\in U$, and

$$\delta(x,X)=\Sigma_{y\in X}\,\delta(x,y),$$

$$\delta^n(x,X)=\Sigma_{y\in X}\,[\delta(x,y)]^n \text{ for } n\in N,$$

$$\overline{\delta}(x,X)=\sum_{y\in X}[\delta(x,y)-\frac{1}{card(X)}\delta(x,X)]^2 .$$

Let us now present an axiomatic approach to representation choice problem. In order to achieve it, let us define several classes of choice functions and show the relationships between these classes. By a representation choice function in a distance space (U,δ) we mean a function: $c\colon \hat{\prod}(U) \to 2^U$.

For $X\in \hat{\prod}(U)$, the set $c(X)$ is called a *representation* of the profile X, where an element of $c(X)$ is called a *representative* (or *consensus*) of the profile X. Let $\boldsymbol{C}$ denote the set of all representation choice functions in space (U,δ).

Definition 2. *A representation choice function $c\in \boldsymbol{C}$ satisfies the postulate of:*

1. *Reliability* (*Re*) *iff*
   $c(X)\neq\emptyset$
2. *Consistency* (*Co*) *iff*
   $(x\in c(X)) \Rightarrow (x\in c(X \dot{\cup} (1*x)))$
3. *Quasi-unanimous* (*Qu*) *iff*
   $(x\notin c(X)) \Rightarrow ((\exists n\in N)(x\in c(X \dot{\cup} (n*x))$
4. *Proportion* (*Pr*) *iff*
   $(X_1\subseteq X_2 \wedge x\in c(X_1) \wedge y\in c(X_2)) \Rightarrow (\delta(x,X_1)\leq\delta(y,X_2))$
5. 1-*Optimality* (*O*$_1$) *iff*
   $(x\in c(X)) \Rightarrow (\delta(x,X)=\min\limits_{y\in U} \delta(y,X))$
6. *n-Optimality* (*O$_n$*) *iff*
   $(x\in c(X)) \Rightarrow (\delta^n(x,X)=\min\limits_{y\in U} \delta^n(y,X))$ *for* $n\geq 2$.

Remark: The above postulates are accepted conditions for representation choice functions. Although they are rather intuitive, an additional clarification can be helpful. The postulate *Reliability* assumes that a representative for each (nonempty) profile should always exist. However, the question of whether the representative is a good one or not for a given profile, requires further conditions. Reliability is a known condition for consensus choice function. The postulate *Consistency* states that if some element x is in a representation of a profile X, then once this element is again added to X, the representation should still contain x. Consistency is a very important requirement for consensus functions, because it enables users to understand a consensus rule's behavior, if the results of separate choices are combined. In the literature some authors have used the consistency defined by Condorcet [1] "... *if two disjoint subsets of voters V and V' would choose the same alternative using* (*social choice function*) *f, then their union should also choose this alternative using f*". Consistency has appeared in every work related to consensus problems. In a broad set-theoretic model for consensus methods has been presented, in which the fundamental role of consistency could be appreciated. For representation choice problems we propose a different definition of consistency, which seems to be more suitable than Condorcet consistency [3],[11] but in this section we will show the relationships between these notions. According to the postulate *Quasi-unanimity*, if an element x is not any representative of a profile X, then it should be a representative of a set X' containing X and n elements x for some n. In other words, each element from U should be chosen as the representative of a profile X if it occurs in X enough times. The postulate *Proportion* is a natural condition because the bigger the profile, the greater is the difference between its representative and its elements. Particular attention should be paid to the two final postulates. The postulate 1-*Optimality* [5],[12],[15] requires the representative to be as near as possible to elements of the profile. The postulate *n-Optimality* for n=2, on the other hand, states that the sum of the squared distances between a representative and the profile's elements should be minimal. Notice that the number $\overline{\delta}$ defined above is a measure of the uniformity of distances between some element x to the elements of profile X. It is an intuitive condition that the representative should not only be nearest to the versions, but should also generate the most uniform distances to them. As it will be stated below, the postulate 2-*Optimality* specifies a representation choice functions which, to a certain degree, satisfy this condition. Notice also that there are other well-known postulates defined in the literature such as *faithful*, *unanimity* [3] or *Condorcet' consistency* [11], but as it will be shown later, they are some interesting consequences of postulates proposed here. The postulates P1-P4 have been shown to be very important properties of the following representation choice functions [15]:

$$c_n(X) = \{x\in U\text{: } \delta^n(x,X) = \min_{y\in U} \delta^n(y,X)\}\text{, and}$$

$$\overline{c}(X) = \{x\in U\text{: } \overline{\delta}(x,X) = \min_{y\in U} \overline{\delta}(y,X)\} \text{ for } X\in \hat{\prod}(U),\ n\in N.$$

These functions and the postulates are analysed in detail in work [15].

In works [4], [10], [14]-[20] a methodology for consensus choice and its applications in solving conflicts in distributed systems is presented. This methodology could be partitioned into 2 parts. In the first part general consensus methods which may effectively serve to solving multi-value conflicts are worked out. For this aim a consensus system, which enables describing multi-value and multi-attribute conflicts is defined and analyzed (it is assumed that the attributes of this system are multi-value). Next the structures of tuples representing the contents of conflicts are defined as distance functions between these tuples. Finally the consensus and the postulates for its choice are defined and analyzed. For defined structures algorithms for consensus determination are worked out. Besides the problems connected with the susceptibility to consensus and the possibility of consensus modification, are also investigated. The second part concerns varied applications of consensus methods in solving of different kinds of conflicts which often occur in distributed systems. The following conflict solutions are presented: reconciling inconsistent temporal data; solving conflicts of the states of agents’ knowledge about the same real world; determining the representation of expert information; creating an uniform version of a faulty situation in a distributed system; resolving the consistency of replicated data and determining optimal interface for user interaction in universal access systems. An additional element of this work is the description of multiagent systems AGWI aiding information retrieval and reconciling in the Web, for which implementation the platform IBM Aglets is used.

3 Representation Reduction Problem

In many practical applications one is interested in achieving a small number of representatives (that is small number of elements of the representation). The reason is simple, representatives enable to make decisions, the smaller the number of them is, the more unambiguous is the final decision.

For reduction needs we generalize the definition of representation choice functions c_n and $\overline{c}$ given in Section 2 by adding an additional argument. That is by a representation choice function we understand the following function:

$$R\colon \hat{\prod}(U) \times \Pi(U) \to 2^U$$

such that $R(X,Y) \subseteq Y$ for all $X \in \hat{\prod}(U)$ and $Y \in \Pi(U)$. Set X is called a profile and set Y is called a choice domain. Thus we have $c_n(X) = R_n(X,U)$ and $\overline{c}(X) = \overline{R}(X,U)$.

Let $\boldsymbol{R} = \{\,\overline{R}\,, R_n$ for n=1,2,…$\}$ be the set of all defined representation choice functions.

The subject of our interests is relied on modification of the profile X and domain Y so that the cardinality of representation $R(X,Y)$ is small. For this aim we accept the following assumption: For each profile X all elements of universe U are distinguishable, that is for each pair $y,z \in U$ $(y \neq z)$ there exists an element $x \in X$ such that $\delta(y,x) \neq \delta(z,x)$. Owing to this assumption each element of U should generate a different set of distances to the elements of the profile.

Following we give an example which should illustrate this problem.

Example 1. From a set of candidates (denoted by symbols A, B, C...) 4 voters have to choose a committee (as a 1 or 2-element subset of the candidates' set). In this aim each of voter votes on such committee which in his opinion is the best one. Assume that the votes are the following: $\{A,B\}$, $\{A,C\}$, $\{B,C\}$ and $\{A\}$. Let the distance between 2 sets of candidates is equal to the cardinality of their symmetrical difference. In this case the universe U is equal to the collection of all 1 or 2-element subsets of candidates' set. Using function R_1 (the most practically applied function) to determine the representation of profile $X=\{\{A,B\}, \{A,C\}, \{B,C\}, \{A\}\}$ from domain Y being the collection of all 1 or 2-element subsets of $\{A,B,C\}$, we obtain the following representatives: $\{A,B\}$, $\{A,C\}$ and $\{A\}$. One can notice that the number of the representatives is equal 3 and thus the choice result is not unambiguous. It is then needed to reduce the representatives number in some way.

Below we present the reduction methods and some their properties.

4 Methods for Reduction

In this section we present some methods for reducing the number of representatives in the representation of a set.

4.1 Method 1: Choice by Additional Criterion

This method is based on using an additional criterion (choice function) to the previous for achieving smaller representation. For example, after making choice for profile X and domain Y on the basis of function R_1 one can use function R_2 for which the domain is not X but $R_1(X,Y)$. Thus the result of choice should be $R_2(X,R_1(X,Y))$. Of course we have: $R_2(X,R_1(X,Y)) \subseteq R_1(X,Y)$. In the most of cases there should be $R_2(X,R_1(X,Y)) \subset R_1(X,Y)$.

By $D(X,y)$ we denote the set of all distances from and element y of domain Y to elements of profile X, that is

$$D(X,y) = \{\delta(x,y): x \in X\}.$$

The following property is true:

Theorem 1. *For each profile* $X \in \hat{\prod}(U)$ *and domain* $Y \in \Pi(U)$ *the following equalities are true:*

a) $R_1(X,R_2(X,Y)) = \overline{R}(X,R_2(X,Y))$

b) $R_2(X,R_1(X,Y)) = \overline{R}(X,R_1(X,Y))$

c) There exists such a natural number n that representation $R_n(X,R_1(X,Y))$ *has exactly* 1 *element if and only if for all* $y,y' \in R_1(X,Y)$, $y \neq y'$ *we have* $D(X,y) \neq D(X,y')$.

The above theorem shows essential properties of choice made by additional criteria. Namely, if in the first step one choose representation by function R_1 and next by function R_2 then the result is the same as in the case when firstly the choice is made by function $\overline{R}$ and next by function R_2. The similar result is in the case when firstly the choice is made by function R_2 and next by function R_1. Finally, if all representatives chosen by function R_1 are distinguishable, then there exist such n that using function R_n as the second criterion gives exactly one representative. This is also true for all other functions used as the first choice criterion.

Example 2: For the profile in Example 1 if in the second step we use function R_2 then the result will contain not 3 but 2 representatives, which are $\{A,B\}$ and $\{A,C\}$. Besides notice that $D(X,\{A,B\}) = D(X,\{A,C\})$, that is representatives $\{A,B\}$ and $\{A,C\}$ are not distinguishable. In this case it is not possible to determine such n that representation $R_n(X,R_1(X,Y))$ has exactly 1 element.

4.2 Method 2: Additional Choice by Adding Representatives to Profiles

In this method we propose the choice in 2 steps: In the first the representation is chosen on the basis of some criterion If the number of representatives is large, then we create a new profile by adding all the representatives to the previous profile, and next to use the same or other criterion to make the second choice. This method is very effective because in the second choice the number of representatives is always equal 1. We have the following:

Theorem 2. *For each profile* $X \in \hat{\prod}(U)$, *domain* $Y \in \Pi(U)$ *and function* $R \in \boldsymbol{R}$ *the number of elements of set* $R((X \dot{\cup} R(X,Y)),Y)$ *is equal* 1.

The following example illustrates the method:

Example 3: From the Example 1 it follows that $R_1(X,Y) = \{\{A,B\}, \{A,C\}, \{A\}\}$. After adding these representatives to the profile we have new profile

$X' = \{\{A,B\}, \{A,B\}, \{A,C\}, \{A,C\}, \{B,C\}, \{A,D\}, \{A\}\},$

for which $R_1(X',Y) = \{A\}$, that is $R_1(X',Y)$ contains exactly 1 representative.

4.3 Method 3: Additional Choice by Reducing Profiles

In this method we propose to reduce the representatives number by moving from the profile those elements which "spoils" its consistency. The problem of measuring consistency of profiles has been analyzed in works [6],[13] in which the authors propose some consistency functions. One of these functions is the following:

$$Co(X) = 1- \frac{\frac{1}{k^2}\sum_{x,y\in X}\delta(x,y)}{\max\limits_{x,y\in U}\delta(x,y)}$$

where k=card(X).

Let $$E(y,X) = \tfrac{1}{k}\sum_{x\in X}\delta(x,y)\,.$$

One can notice that

$$Co(X) = 1 - \frac{\frac{1}{k}\sum_{y\in X}E(y,X)}{\max\limits_{x,y\in U}\delta(x,y)}\,.$$

Thus the element x of profile X for which $E(x,X) = \max\{E(y,X)\colon y\in X\}$ should most "spoil" its consistency. That is if we remove this element from X then the consistency of new profile should increase.

The reduction method we propose in this subsection is based on several steps of representation choice. At the first step, we make the choice on the basis of function R_1. If the number of representatives is large in the second step we remove from the profile the element which most "spoils" the consistency of X and make the second choice for this profile using still function R_1. And so on, the process should be stopped when the number of representatives is equal to 1. The following example should illustrate this method.

Example 4: With the same assumptions from Example 1, let us consider the profile $X=\{\{A,B\}, \{C\}, \{B,C\}, \{A,C\}, \{A\}, \{D\}\}$. We can notice that the representation choice using function R_1 gives 3 representatives: $\{A\ \}$, $\{A,C\}$ and $\{C\}$. For reducing this number we notice that the last vote, that is $\{D\}$, most spoils the profile consistency because the value $E(\{D\},X)$ is maximal. After removing this element, we have the following profile $X' = \{\{A,B\}, \{C\}, \{B,C\}, \{A,C\}, \{A\}\}$, for which the representation choice on the basis of function R_1 give only one representative $\{A,C\}$.

5 Conclusions

In this paper 3 methods for solving the problem of representation reduction are presented. It contains also some results of the analysis of these methods. These results enable to choose a proper method dependently from real situations. The contents of the algorithms for realizing this task are rather simple but their computation complexity is in large degree dependent from the complexity of algorithms for choice using particular functions. The future works should concern the analysis of the consequences of applications of these methods. This analysis should show the relationships between a representative after reduction and a representative before performing this process. The results of this work should be useful in designing and implementing intelligent user interfaces [19].

References

1. Arrow K.J.: Social choice and individual values. Wiley New York (1963).
2. Aizerman M.A.: New problems in the general choice theory. Social Choice Welfare **2** (1985) 235-382.
3. Barthelemy J.P., Janowitz M.F.: A formal theory of consensus. SIAM J. Discrete Math. **4** (1991) 305-322.
4. Daniłowicz C., Nguyen N.T.: Consensus methods for solving inconsistency of replicated data in distributed systems. Distributed and Parallel Databases – An International Journal **14**(1) (2003) 53-69.
5. Daniłowicz C., Nguyen N.T.: Criteria and functions for expert information representation choice. In: Kłopotek et al. (eds), Advances in Soft Computing, Proceedings of 10th Int. Conference on Intelligent Information Systems'2001, Physica-Verlag (2001) 225-234.
6. Daniłowicz C., Nguyen N.T., Jankowski L.: Methods for choice of representation of agent knowledge states in multiagent systems. Monograph, Wroclaw University of Technology Press (2002) (in Polish)
7. Day W.H.E.: Consensus methods as tools for data analysis. In: Bock, H.H. (ed.): Classification and Related Methods for Data Analysis. North-Holland (1988) 312-324.
8. Fishburn P.C.: Condorcet social choice functions. SIAM J. App. Math. **33** (1977) 469-489.
9. Katarzyniak R., Nguyen N.T.: Solving conflicts of agent knowledge states in multiagent systems. In: Proceedings of 29th SOFSEM. Lecture Notes in Computer Science **2540** (2002) 231-239.
10. Katarzyniak R., Nguyen N.T.: Modification of Weights of Conflict Profile's Elements and Dependencies of Attributes in Consensus Model. Lecture Notes in Artificial Intelligence **2475** (2002) 131-138.
11. McMorris F.R., Powers R.C. (1995): The median procedure in a formal theory of consensus, SIAM J. Discrete Math. 14, 507-516.
12. McMorris F.R., Powers R.C., The median function on weak hierarchies. DIMACS Series in Discrete Mathematics and Theoretical Computer Science **37** (1997) 265-269.
13. Malowiecki, M., Nguyen, N.T., Zgrzywa, M.: Using Consistency Measures and Attribute Dependencies for Solving Conflicts in Adaptive Systems. To appear in: Proceedings of ICCS 2004, Krakow Poland June 2004, Lecture Notes in Computer Science (Springer-Verlag).
14. Nguyen, N.T.: Using Consensus Methods for Solving Conflicts of Data in Distributed Systems. In: Proceedings of 27th SOFSEM. Lecture Notes in Computer Science **1963** (2000) 409-417.
15. Nguyen N.T.: Using distance functions to solve representation choice problems. Fundamenta Informaticae **48**(4) (2001) 295-314.

16. Nguyen N.T.: Consensus system for solving conflicts in distributed systems. Journal of Information Sciences **147** (2002) 91-122.
17. Nguyen N.T.: Representation Choice Methods as The Tool for Solving Uncertainty in Distributed Temporal Database Systems with Indeterminate Valid Time. In: Proceedings of IEA/AIE-2001, Lecture Notes in Artificial Intelligence **2070** (2001) 445–454.
18. Nguyen N.T.: Using consensus for solving conflict situations in fault-tolerant distributed systems, in: Proceedings of First IEEE/ACM Symposium on Cluster Computing and the Grid 2001, Australia, IEEE Computer Press (2001) 379-385.
19. Nguyen N.T., Sobecki J.: Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems. Journal of Universal Access in the Information Society **2**(4) (2003)342-358 (Springer-Verlag).
20. Nguyen N.T.: Consensus-based timestamps in distributed temporal databases. The Computer Journal 44(5) (2001) 398-409.

Incremental Maintenance of All-Nearest Neighbors Based on Road Network

Jun Feng[1, 2], Naoto Mukai[3], and Toyohide Watanabe[3]

[1] Department of Information Engineering, Graduate School of Engineering,
Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan
feng@watanabe.nuie.nagoya-u.ac.jp

[2] College of Computer and Information Engineering, Hohai University,
1 Xikang Road, Nanjing, Jiangsu 210098, China

[3] Department of Systems and Social Informatics, Graduate School of Information Science,
Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan
{naoto,watanabe}@watanabe.nuie.nagoya-u.ac.jp

Abstract. In this paper, we propose a method to incrementally maintain all-nearest neighbors on road network. The nearest neighbor on road network is computed based on the path length between a pair of objects. To achieve an efficient maintenance process of all-nearest neighbors on road network, the cost of path search should be decreased. A method is proposed for efficient path search by minimizing the search region based on the previous results and heuristics.

1 Introduction

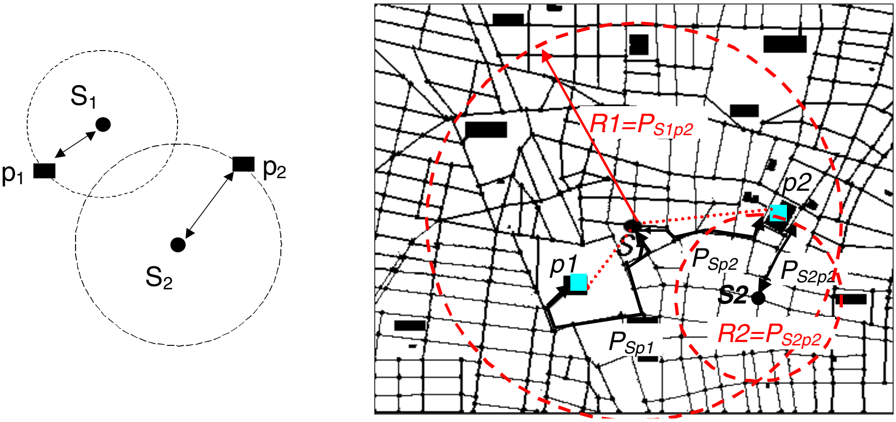
There are many researches to compute all-nearest neighbors (ANN) for spatial databases [1]. This work can be regarded as a kind of spatial join. The spatial join operation is similar to the join operation in relational databases. It is defined on two sets of objects, and computes a subset of the Cartesian product of two sets, determined by a spatial predicate, which prescribes a certain spatial relationship between objects in the result.

For example, there are a source dataset S and a target dataset P. To compute the all-nearest neighbors for S means that to do distance semi-join with S and P, results in finding the nearest neighbors (NN) in P for all objects in S. The distance semi-join can be implemented using a NN algorithm: for each object in S, NN computation is performed in P, and the resulting array of distances is sorted once all neighbors have been computed. However, how to maintain the result when there is a change of the two sets was seldom discussed: e.g., there is a new object added to P. This maintenance work is needed especially when the semi-join has been done on an underlying spatial or geographical distance.

In this paper, we center on the maintenance method of ANN result set based on an underlying geographical distance: the path length between objects on the road net

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 164-169, 2004.

work. On road network, the nearest neighbor based on the path length between objects may be different from that based on the straight-line distance. We give an example of road network in Fig.1: *s1* and *s2* are objects in the source dataset S, and *p1* and *p2* are objects in the target dataset P. It can be directly observed that the target object *p1* is nearer to *s1* than *p2* on the straight-line distance: however, by comparing the path lengths among them, the nearest neighbor for *s1* is *p2*, and that for *s2* is also *p2*. Therefore, ANN is a set {(*s1*, *p2*, P_{s1p2}), (*s2*, *p2*, P_{s2p2})}, where P_{sipj} means the path length between si and pj.



(1) (2)

Fig. 1. All-nearest neighbors of sets S and P are different when there is an underlying geographic distance. (1) ANN on 2D space; (2) ANN on road network.

The computation of ANN on road network is a special spatial join which relates to not only the spatial attributes of objects in the two datasets, but also relates to the situations of road networks which these objects are located on. Because any spatial structure is implemented based on the spatial attributes of objects, the path length based on a network between spatial objects is regarded as an inside property of objects which is independent of the spatial attributes in spatial structures. Therefore, to take advantages of spatial structure, how to bridge the inside property of object with the spatial attributes is the main point of this paper.

2 Propositions for Bridging NN Search on 2D Space and Road Network

There are many spatial data structures: typical one is R-tree [2], proposed for efficient accessing to spatial dataset. And also many R-tree-based methods for solving NN queries have been proposed [3, 4]. However, because R-tree is created based on the coordinates of spatial objects, in 2D space, NN search based on R-tree uses straight-line distances among objects as a standard, actually. While NN search on road network should use path length as a standard, which is not only decided by spatial relations but also the connection relations among objects [5]. Considering the high efficiency of R-tree for spatial queries, we give two propositions for bridging the straight-line distance and the path length on road network, which lead to an efficient NN search on road network.

[Proposition 1.] *For a source point S and a target object p, when the path length from S to p is r, if any target object is nearer to S than r, it can only be found inside a circle region, denoted as r-region, whose center is S and where radius is r (the circular area in Fig. 2(1)).*

We leave the proof out in this paper, as it comes from the fact that any road segment outside *r-region* can only lead to a path longer than *r* from S.

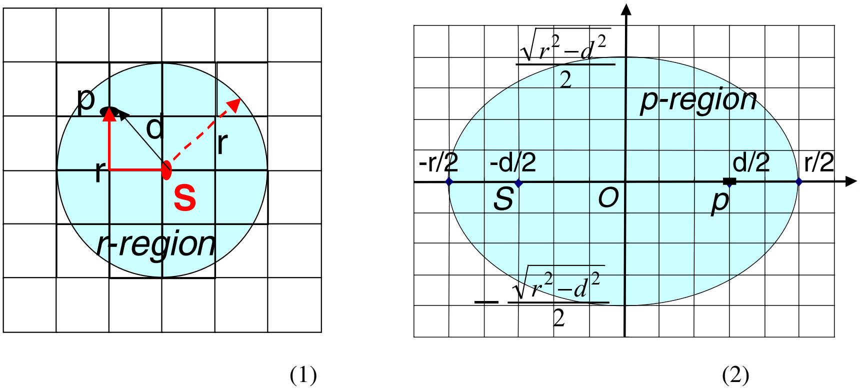


Fig.2. Search regions generated based on straight-line distance of objects on 2D space and path length between objects on road network. (1) *r-region* for NN search on road network; (2) *p-region* for path search on road network.

[Proposition 2.] *For two points S and p on the road network with straight-line distance d, to test whether there is a path shorter than r from S to p can be based on a path search region, denoted as p-region. Inside p-region, the sum of both the straight-line distance between any nodes and S and that between this node and p is not longer than r (the elliptic area in Fig. 2(2)).*

In Fig.2 (2), for an easy description we define the coordination for them. The origin O is on the center of line *sp*, the x-axis passes along line *sp*, and the y-axis is perpendicular to the x-axis on the origin O. *p-region* is defined as:

$$p - region(d,r) = \{(x,y) \mid \sqrt{(x+d/2)^2 + y^2} + \sqrt{(x-d/2)^2 + y^2} \le r\}. \tag{1}$$

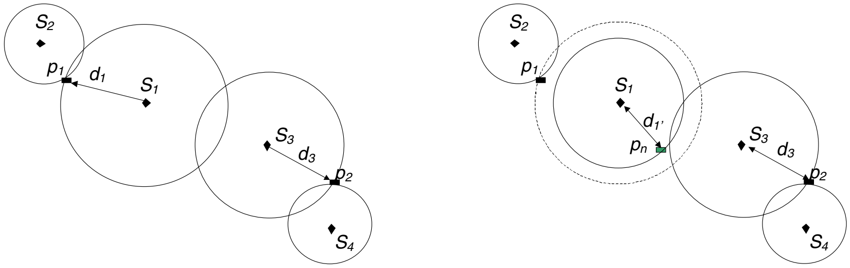
This means that when the straight-line distance between *s* and *p* is *d*, if there is any path shorter than *r* from *s* to *p*, all the road segments on this path could only be found inside *p-region (d, r)*. This proposition comes from the fact that any road segments outside *p-region* can only result in a path longer than *r* from *s* to p.

3 Maintenance of All-Nearest Neighbors on Road Network

To achieve an efficient maintenance of all nearest neighbors on road network, we first observe ANN set on 2D space and that on road network.

ANN set on 2D space can be regarded as a set of circle regions, which are generated for every object in S (see Fig. 3 (1)). Here, we call these circles as NN-regions for S. Assuming that all the objects in S and P are indexed by R-tree, respectively, when a new target object p_n is inserted into P, whether the insertion of p_n leads to a modification to ANN will be tested:

1) Find out all NN-regions which p_n is located in. We call these regions as influence regions;
2) Compute the straight-line distance between p_n and all the center objects of influence regions;
3) Modify the radii of these influence regions with the new distance, and set p_n as new NN. The modified ANN is depicted in Fig.3 (2), where NN-region for s1, (s_1, p_1, d_1), is an influence region by inserting p_n: p_n is new NN for $s_{1,}$ and NN-region of s_1 is changed to $(s_1, p_1, d_{1'})$. There is only one influence region of p_n, so the maintenance of ANN is completed.



(1) (2)

Fig. 3. NN-regions for ANN set on 2D space. (1) ANN for S and P; (2) new ANN generated after p_n has inserted into P.

ANN on road network also generates a region for every object in S: however, the region is *r-region* defined in the previous section, because the path length between objects decides NN. Therefore, when p_n is inserted into P, the modification to ANN should also be based on the path length.

In Fig. 4, we depict the situation of ANN on road network. Being different from the situation of 2D space, when p_n is inserted into P, not all the *r-region*s which p_n is located in are modified with new NN. Though the modification can be started from searching the *r-region*s which p_n is located in, the following test steps should be based on the road network—to compute the path length between p_n and those centers of influence *r-region*s. If new path is shorter than the old (radius of *r-region*: *r*), p_n is the new NN, and *r-region* is modified*;* otherwise, no modification should be done to ANN. Here, the centers of influence *r-region*s are S_1 and S_3, therefore, the paths from S_1 *to* p_n and S_1 *to* p_n are computed. However, only the *r-region* of S_3 is modified with the new radius. Some algorithms, such as Dijkstra's algorithm [6], can realize the path search. However, the cost of path search is related to the scale of road network. When the scale is large, in other words when there are a great number of road segments in the road network, the path search would be a cost process.

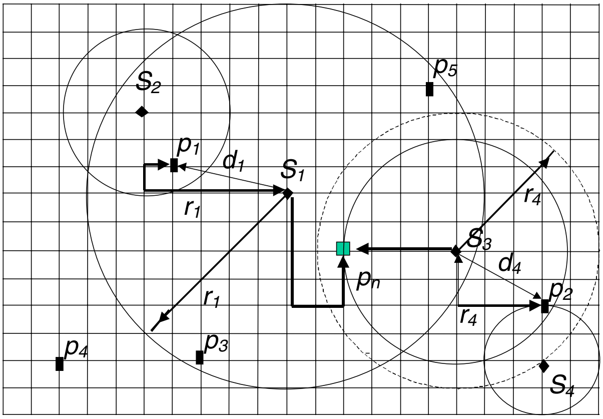


Fig. 4. NN-regions for ANN set on road network: modification of ANN results in NN-region modification when a new p is inserted into P.

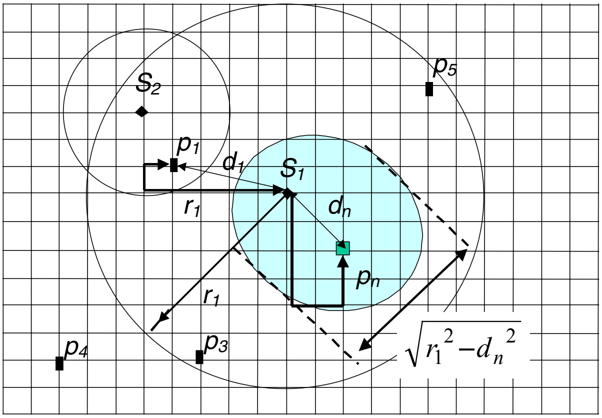


Fig. 5. *p-region* for testing new NN on road network.

To solve this problem, we adopt *p-region* (defined in Section 2) for an efficient path search. Here, we generated *p-region (d, r)* for an influence *r-region*, where *d* is the straight-line distance between p_n and the center of influence *r-region*, and *r* is the radius of the influence *r-region*.

In Fig.5, we give an example of *p-region*. Because p_n can be set as the new NN for S_1 only when there is a path shorter than r_1, the path search from S_1 to p_n only needs to be done inside *p-region*. *p-region* is an ellipse decided by r_1 and the two nodes S_1 and

p_n. Here, the area of *p-region* (d_n, r_1) can be approximated as formula (2) and the relation between areas of *p-region* and *r-region* can be computed as formula (3).

$$Area(p-region) \propto r_1 \times \sqrt{r_1^{\ 2} - d_n^{\ 2}}\,. \tag{2}$$

$$\frac{Area(p-region)}{Area(r-region)} \propto \frac{r_1 \times \sqrt{r_1^{\ 2} - d_n^{\ 2}}}{\pi r_1^{\ 2}} \leq \frac{\sqrt{r_1^{\ 2} - d_n^{\ 2}}}{\pi r_1} < \frac{1}{\pi}\,. \tag{3}$$

As the path search can be limited to a smaller subset of road segments inside *p-region*, the cost is quite decreased. The maintenance of ANN on road network can also be realized efficiently.

4 Conclusion

In this paper, we defined search regions in ANN maintenance process. *R-region* is used to filter the candidate for inside attribute test—shortest path search, and *p-region* is used to test the inside attribute of spatial objects. The maintenance of ANN based on road network can be realized efficiently. Moreover, our method can be used to other kinds of ANN maintenances, which are based on some underlying distances different from the straight-line distance but with some relations to it.

References

1. Hjaltason, G.R.: Incremental distance join algorithms for spatial databases, Proc. Of ACM SIGMOD'98 (1998) 237–248
2. Guttman, A.: R-Trees: A dynamic index structure for spatial searching, Proc. of ACM SIGMOD'84 (1984) 47–57
3. Tao, Y.F., Papadias, D., Shen Q.M.: Continuous Nearest Neighbor Search, Proc. of VLDB'02 (2002) 287-298
4. Song, Z.X., Roussopoulos, N.: K-Nearest Neighbor Search for Moving Query Point, Proc. of SSTD'01 (2001) 79-96
5. Feng, J., Watanabe, T.: A Fast Search Method of Nearest Target Object in Road Networks, Journal of the ISCIE, Vol. 16, No. 9 (2003) 484-491
6. Christofides, N.: Graph Theory: An Algorithmic Approach, Academic Press Inc. (London) Ltd. (1975).

Knowledge Intensive Interpretation of Signal Data

Keith Mason and Catherine Howard

Electronic Warfare and Radar Division,
Systems Sciences Laboratory,
Defence Science and Technology Organisation,
PO Box 1500 Edinburgh SA 5111,
Australia
{keith.mason, catherine.howard}@dsto.defence.gov.au

Abstract. This paper discusses the use of knowledge for the interpretation of signal data in a laboratory prototype system known as ELEXSA (ELINT Exploitation for Situation Awareness). ELEXSA uses knowledge intensive techniques to yield information products that aim to assist a military commander to comprehend the significance of detected entities, their relationships with own force platforms, and the effects of those relationships relative to the commander's goals. The core interpretation components of ELEXSA are structured in the manner of the λJDL information fusion model. These components make intensive use of corporate knowledge bases detailing the capabilities of military systems, models of operational concepts and of the physics of propagation of radar signals. An illustration is provided which shows how an ELEXSA information product is constructed. Technical issues are identified and future directions of this work are also presented.

1 Introduction

The detection and analysis by passive sensing systems of radar signals emitted by military platforms (vessels, aircraft and land-based systems) provides an important source of surveillance data that is known as ELINT (Electronic Intelligence [1]). The output of such sensors typically includes the parameters of detected emissions and the location of the radar emitter. Interpretation of this data is usually required for it to be used by a military commander.

For example, in a military conflict, adversary radars that are associated with weapon systems can pose a lethal threat to own force platforms. Planning a safe passage through a radar protected region or planning a mission to disable a threat radar site requires understanding of the capabilities of threat systems and their relationship to the characteristics and capabilities of own force platforms. Obtaining this understanding generally requires expertise in signal interpretation as well as knowledge of military systems. While military commanders are expert in tactics and military systems, they may only have limited skills for interpreting signal data. This paper discusses a laboratory prototype system known as ELEXSA (ELINT Exploitation for Situation Awareness) which is being developed by Australia's Defence Science and Technology Organisation (DSTO).

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 170-175, 2004.

ELEXSA's inputs are parametric descriptors of radar signals intercepted by passive sensors. ELEXSA uses knowledge intensive techniques to yield information products that aim to assist a military commander to comprehend the significance of detected entities, their relationships with own force platforms, and the effects of those relationships relative to the commander's goals.

Section 2 presents the information architecture supporting ELEXSA. Section 3 describes the ELEXSA components for signal interpretation and Section 4 outlines technical issues and areas for further research.

2 ELEXSA and EXC3ITE

The Experimental Command, Control, Communications and Intelligence Technology Environment (EXC3ITE) is a DSTO demonstrator aiming to develop and leverage the use of middleware in distributed information systems. The benefits of middleware architectures include evolvability, reuse, scalability, and reliability, and thereby offers significant cost savings in developing a large information infrastructure.

The three layer information architecture developed in EXC3ITE shown in Figure 1 provides the framework for ELEXSA. The Data Layer includes connectivity to real and synthetic data sources. The ELEXSA interpretation components fall within the analysis services of the Business Logic Layer. The Visualisation Layer includes web based services that enable ELEXSA's human-computer interface.

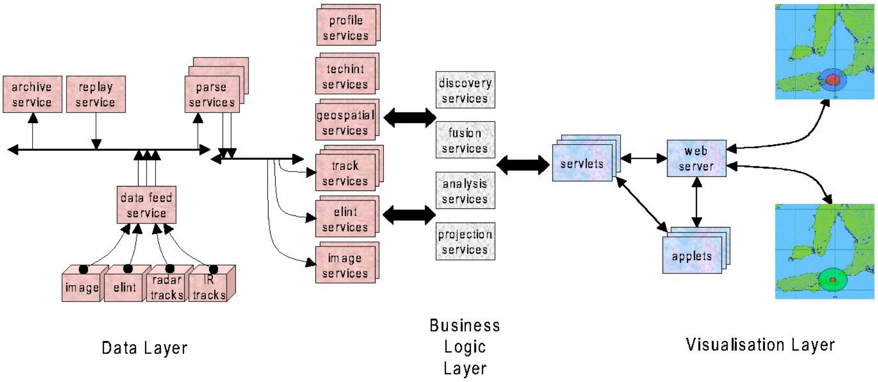


Fig. 1. The Information Architecture

ELEXSA has been successfully demonstrated to yield near real-time information products derived from radar signals intercepted by distributed uninhabited aerial vehicles [2]. Future work may extend the ELEXSA concept to include other business logic functions indicated in Figure 1, such as fusing data from a variety of sensor types.

3 The Business Logic

The core interpretation components of ELEXSA are structured in the manner of the λJDL [3] information fusion model. λJDL comprises three levels of abstraction: Object, Situation and Impact. These levels incorporate different entities and computation processes. At each level computational processes (termed fusion even if there is only one kind of sensor data) yield assessments. *Object Fusion* is the process of utilising one or more data sources over time to assemble representations of the objects of interest in an area of interest and yields *Object Assessments. Situation Fusion* is the process of utilising one or more data sources over time to assemble representations of the relations of interest between objects of interest in an area of interest and yields *Situation Assessments. Impact Fusion* is the process of utilising one or more data sources over time to assemble representations of the effects of the situations in the area of interest, relative to own force goals and yields *Impact Assessments.*

3.1 Object Assessments

ELEXSA develops Object Assessments through a data driven sequence of processes stimulated by the arrival of signals. These processes include: Emitter Creation or Update, Emitter and Platform Identity Candidate Generation, Enrichment and Pruning.

Emitter and Platform Identity Candidates are generated by matching signal parameters with entries in corporate databases. The Enrichment process fills in details in the candidate platform data structure using values in databases cataloging military capabilities. Typical attributes sought during enrichment include: co-located emitters, mobility limits, radar cross section and weapon lethality details. The Pruning phase matches the Enriched Platform Identity Candidate descriptors against heuristic rules to remove inadmissible candidates, for example candidates with sovereignty that are deemed inconsistent with the detected location.

If more than one platform is postulated at a location, they are only merged into a single entity if there is knowledge indicating that the aggregation is realistic, such as that the adversary does co-locate the particular identified emitters. In the case of an ambiguity that cannot be resolved, the current ELEXSA implementation continues with a set of platform candidates and can provide "worst case" values if required.

The enriched platform descriptor forms the Object Assessment. Visualisation of Object Assessment attributes, such as lethality ranges, aim to assist a military commander to comprehend the significance of detected entities.

3.2 Situation Assessments

ELEXSA develops Situation Assessments through a goal driven sequence of processes initiated by the commander, or by computations deriving an impact assessment. ELEXSA currently evaluates the following situations:

- Detection range by adversary radar systems of own force platforms
- Lethal capability of a protected site against an own force platform

- Lethal capability of an own force platform against an adversary site
- Weapon launch range of own force platform against a protected site

Evaluation of the detection range, for example, requires the use of data such as the radiated power of the threat radar that was derived during enrichment of sensed data, as well as values such as own force platforms radar cross section, which needs to be found in a knowledge base. These derived values comprise a Situation Assessment. Visualisation of Situation Assessments such as depicting the range that a threat radar can detect an own force platform aims to assist a military commander to comprehend the relationship between detected entities and own force platforms.

3.3 Impact Assessments

ELEXSA develops Impact Assessments through a goal driven sequence of processes initiated by the commander. Currently ELEXSA supports the generation of the following Impact Assessments:

- Protection of a strike platform by a jamming mission
- Safe route
- Covert route

Computation of Impact Assessments can require Object Assessment attributes, the evaluation of Situation Assessments, information that can be supplied by the commander or data that has to be found in the information architecture.

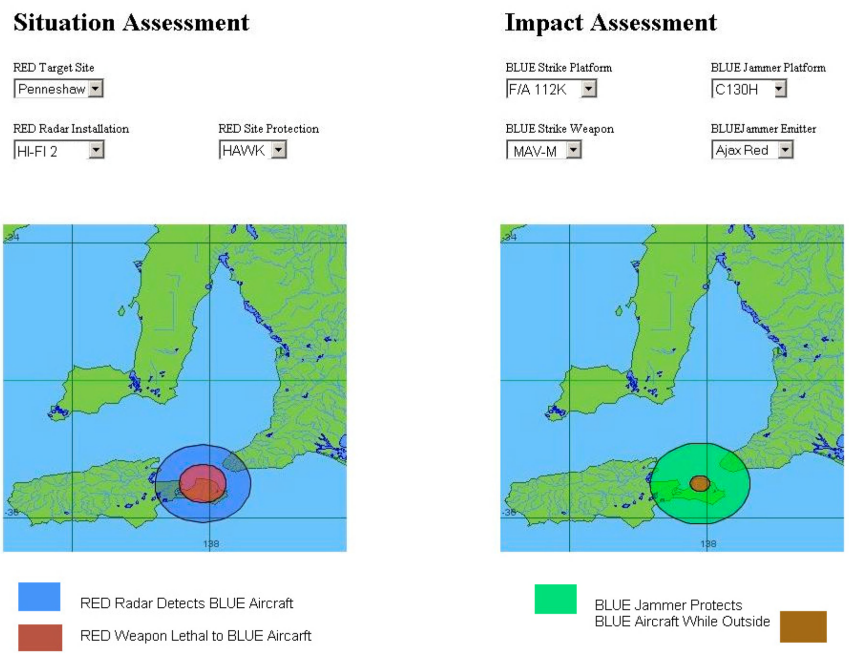
For example, to plan an airborne strike mission against a threat radar site, ELEXSA would need to know the available strike aircraft, the weapons that can be launched by the aircraft, the range at which the weapon should be released and the maximum lethality range of weapon systems associated with the threat radar site.

If the strike platform needs jamming protection for safe ingress to disable the threat radar site, ELEXSA would also need to know the available jamming aircraft, the jammers capable of jamming the threat radar system which can be carried by the jamming aircraft, the range at which the jamming aircraft would be effective against the threat radar during each stage of the mission and the possible effects of terrain shielding.

The range at which the jamming aircraft would be effective against the threat radar can be computed from the radar cross section of the strike platform, the weapon launch range of the strike platform (that is, point of closet ingress), the radiated power of the adversary radar, the signal processing strategy of the adversary radar, the radiated power of the jamming emitter, the jamming strategy, and the physics of radar propagation.

Figure 2 shows an ELEXSA visualization of an Impact Assessment for this jamming goal and associated Situation Assessment. The computed values are also displayed on the screen for an operator to verify system performance.

Visualisation of Impact Assessment aims to assist a military commander to comprehend the effects of the relationships between own and adversary platforms relative to the commander's goals.



Situation Assessment

RED Target Site: Penneshaw

RED Radar Installation: HI-FI 2

RED Site Protection: HAWK

Impact Assessment

BLUE Strike Platform: F/A 112K

BLUE Jammer Platform: C130H

BLUE Strike Weapon: MAV-M

BLUEJammer Emitter: Ajax Red

-34
-35
138
-34
-35
138
RED Radar Detects BLUE Aircraft
RED Weapon Lethal to BLUE Aircarft
BLUE Jammer Protects
BLUE Aircraft While Outside

Fig. 2. An ELEXSA Visualisation

4 Issues and Further Research

This paper has detailed an information architecture and computational strategies for computing object, situation and impact assessments. These computational strategies use knowledge available within the information architecture. However, these strategies also have to know how to find such knowledge, and having located it, how to interpret it for appropriate use within the strategy. The information architecture shown in Figure 1 does include discovery and profile services. Implementation of such services requires a degree of sophistication in the structure of compliant knowledge bases. The corporate knowledge sources used by ELEXSA are in general legacy systems that lack such sophistication. Further work is required in the integration of legacy systems for the benefits identified in Section 2 to be fully realized.

The current data processing boundary between ELEXSA and sensing systems makes the sensing activity responsible for all the signal measurement activities and ELEXSA for all information processing activities. To maximize the synergies between ELEXSA and DSTO activities in distributed sensing systems [4] this boundary could be moved towards the sensor, with ELEXSA performing some signal measurement functions for fusing intercepts from distributed sensors. Research is required not only in the combination of such data (which may arrive with significant latencies), but also in how ELEXSA could exploit the concept of data pedigree in the management of uncertainty and ambiguity in product formation.

To further the ELEXSA concept, a wider breadth of situation and impact assessments is needed. This involves further knowledge elicitation, representation and formation of computational strategies. It may also involve fusing multiple kinds of sensor data.

A desired output of further research is a careful evaluation of how well ELEXSA's visualizations enable a military commander to comprehend the significance of detected entities, their relationships with own force platforms, and the effects of those relationships relative to the commander's goals.

Acknowledgement. The author would like to acknowledge the work of Mark Nelson, Tim Cawley and Holger Kohler of DSTO's Information Sciences Laboratory in developing the EXC3ITE Surveillance Data Services infrastructure that supports ELEXSA.

References

1. Wiley, R.G.: Electronic Intelligence: The Interception of Radar Signals. Artech House. ISBN 0-89006-138-6 (1985)
2. Howard, C., Mason, K.: Force Level Electronic Warfare And Situation Awareness For Future Littoral Warfare. Proceedings of the Land Warfare Conference 2002, Brisbane, ISBN 0 642 70552 6, 47-53 (2002)
3. Lambert, D.: Grand Challenges of Information, Proceedings of Information Fusion 2003, Cairns, 213-220 (2003)
4. Finn, A., Lindsay, T., Brown, K.: Miniature UAV's & Future Electronic Warfare. Proceedings of the Land Warfare Conference 2002, Brisbane, ISBN 0 642 70552 6, 93-107 (2002)

Coalition Formation among Agents in Complex Problems Based on a Combinatorial Auction Perspective

Hiromitsu Hattori, Tadachika Ozono, Takayuki Ito, and Toramatsu Shintani

Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 JAPAN
{hatto, ozono, itota, tora}@ics.nitech.ac.jp
http://www-toralab.ics.nitech.ac.jp/~hatto/Eng/index.html

Abstract. In this paper, we try to apply a combinatorial auction protocol to a coalition formation among agents to solve complex problems. This paper focuses on a scheduling problem that considers various constraints as a complex problem. Constraints on scheduling can be expressed as combinations of items (time slots) in a combinatorial auction. Agents bid for necessary combinations of time slots to satisfy users' requirements. We formalize a combinatorial auction for scheduling as an MIP (Mixed Integer Programming) problem, which integrates the constraints on items and bids to express complex problems. We have experimentally confirmed that our method can obtain a socially preferable schedule in practical time.

1 Introduction

Auctions have been studied in the field of Electronic Commerce (EC). Various studies on auctions have already been made, and many protocols and methods have been developed [2, 3, 5, 9]. The purpose of our work is to apply auction protocols and methods developed in EC studies to a coalition formation among agents.

We focus on a scheduling problem considering various constraints as a complex problem. We are currently attempting to construct a scheduling system based on a multiagent. In our system, each agent makes bids based on user's preferences on events. Agents must determine a consistent schedule by resolving conflicts among agents' preferences expressed by bids. To construct the scheduling systems, we must propose appropriate problem formalization and methods for finding an optimal solution. In this paper, we formalize a scheduling problem as a combinatorial auction. An appropriate schedule can be obtained by solving the winner determination problem in a combinatorial auction. A combinatorial auction protocol [11] is one of the most notable auction protocols for dealing with preferences over multiple items. The winner determination problem is one of determining the optimal allocation of items that can maximize the auctioneer's revenue. Since the winner determination problem is a complicated optimization problem, it can be re-defined as a scheduling problem.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 176-187, 2004.

When a scheduling problem is formalized as a Constraint Satisfaction Problem [13], we need particular methods to relax over-constrained problems. On the other hand, because of the formalization as a combinatorial auction, we can obtain an appropriate solution according to an economical rationality without having to use such particular methods.

The basic idea behind our formalization is that a scheduling is compared to an auction for winning time slots. This perspective is intuitively and easily understandable for users. In our formalization, a schedule can be represented as combinations of items (time slots). Agents bid for necessary combinations of time slots to satisfy users' requirements. In this paper, we deal with various constraints, e.g., the date and time, participants, the order of events, and the interval of events. The greater the variations of bids considering some constraints becomes, the more time-consuming the computation time for finding solution becomes. Nevertheless, decreasing the number of bids reduces the expressiveness of representing a scheduling problem. Therefore, we formalize a combinatorial auction for scheduling as an MIP (Mixed Integer Programming) problem, which integrates the constraints on items and bids to express problems. This integration solves the trade-off between the computation time to find a solution for a combinatorial auction and the expressiveness to represent a scheduling problem.

A combinatorial auction is appropriate for adjustment of time slots. Considering each time slot as an item, we can use a combinatorial auction protocol to effectively deal with events, each of which needs sequential multiple time slots. Without the dealing with combination of items, an agent might obtain multiple time slots that do not increase his/her utility by obtaining them simultaneously. For example, [8] uses a combinatorial auction protocol to determine arrival and departure times of airplanes. In this work, each time needs sequential multiple time slots. Because a combinatorial auction allows complementary preferences over multiple time slots to be explicitly represented, i.e., sequential time slots are worthy and distributed ones are worthless, time slots can be efficiently allocated to each airplane.

The rest of this paper is organized as follows. In Section 2, we describe the outline of a combinatorial auction. In Section 3, we propose our basic formalization that represents a scheduling problem as a combinatorial auction. In Section 4, we describe the formalization based on an MIP. In Section 5, we experimentally evaluate a scheduling based on our formalization. We discuss related work in Section 6, and finally, we make some concluding remarks.

2 Combinatorial Auctions

In a combinatorial auction [11], bidders can bid on combinations of items. The items are assigned in order to maximize the auctioneer's revenue. A combinatorial auction protocol can increase both the bidders' utility and the auctioneer's utility.

To find a revenue maximizing allocation, the auctioneer must solve the winner determination problem (i.e., the problem of determining what items each bidder gets).

Let G denotes a set of items and A denotes a set of bidders. The highest bid for a combination of items S is defined as:

$$\bar{b}(S) = \max_{i \in A} b_i(S)$$

where $b_i(S)$ represents bidder i's bidding on the combination $S \subseteq G$. Then, the optimal allocation of items is the solution of the following formula:

$$\arg\max_{\chi} \sum_{S \in \chi} \bar{b}(S)$$

where χ is defined as follows:

$$\chi = \{S \subseteq G \mid S \cap S' = \varnothing \text{ for every } S, S' \in \chi\}$$

Namely, χ is a set of allocations of items and an identical item never been in combinations included in χ.

3 Solving a Scheduling Problem as a Combinatorial Auction

3.1 Definition of Scheduling Problem

We deal with a scheduling problem which consists of several events (e.g., meeting scheduling, computer resource sharing). During the scheduling, it is necessary to adjust each agent's schedule, i.e., agents must stagger the start and the end time of events or cancel some of them. For a scheduling, each agent declares constraints and a valuation for each event. We consider three constraints on each event, (1) a list of participants, (2) the length, and (3) the duration of an event. Namely, the start time of an event is not be fixed.

In this paper, each participant and resource (e.g., meeting room, classroom) is represented as r, and a set of them is represented as $R = \{r_1, r_2, ...\}$. For simplicity, the word "resource" is used to represent a participant and resource. The schedule of each resource consists of some time slots, each of which is a unit of the schedule. For example, when we assume one time slot means one hour in the real world, one day is represented as a set of 24 time slots. Each time slot is denoted as t_i. The time goes by in the increasing order of i, that is t_1 is preceded by t_2. The j-th time slot of resource r_i is denoted by t_{ij}. Let T_j be a set of j-th time slots for all resources and $T_{j\{r1,r2,...,rk\}}$ be a set of j-th time slots of resources r_1, r_2, ..., r_k. Fixing event means that a certain agent, which is a host of an event, wins some required time slots. Assuming that an agent wants to hold a meeting with three other agents from 14:00 to 15:00, the host must win one time slot "14:00-15:00" of the others and a meeting room simultaneously. That is to say, winning time slot t_{ij} means to purchasing the right to restrict r_i during a duration of time t_{ij}. r_i's schedule E_i is denoted as follows:

$$E_i = \{e_{i1}, e_{i2}, \ldots, e_{in}\} \quad (n \geq 0)$$
$$e_{ij} = (C_{ij}, R_{ij}, v_{ij})$$

where e_{ij} denotes each event and is distinguished by parameters, C_{ij}, R_{ij}, and v_{ij}. C_{ij} is a set of constraints on time and is represented using logic symbols; it includes the start time T_{ij}^0, the end time T_{ij}^1, and the length of the event, $T_{ij}^1 - T_{ij}^0$. For example, when the start time is between t_a and t_b, the end time is between t_c and t_d, and the length is between l_0 and l_1, the constraint on time is represented as $C_{ij} = (t_a \le T^0_{ij} \le t_b) \wedge (t_c \le T^1_{ij} \le t_d) \wedge (l_0 \le T^1_{ij} - T^0_{ij} \le l_1)$. R_{ij} denotes a set of resources for e_{ij}; it is a subset of R, i.e., $R_{ij} \subseteq R$. v_{ij} is the valuation for e_{ij}.

In addition to constraints of each event, we deal with constraints among multiple events. To put it concretely, "**number constraint**" and "**order constraint**" are introduced. The number constraint is for selecting m events from n events. Given that E is a set of all events, the number constraint is represented as $|E'_0| = m$, $|E_0| = n$, and $E'_0 \subseteq E_0 \subseteq E$. This constraint enables exclusivity among events to be expressed. The order constraint is for specifying an order and an interval between two events. For example, the order constraint $e_{i1} <_l e_{i2}$ means that the end time of e_{i1} is l slots ahead of the start time of e_{i2}.

3.2 Formalization of a Scheduling as a Combinatorial Auction

In this section, we formalize a scheduling problem described in Section 3.1 as a combinatorial auction. A time slot is regarded as an item in an auction. A bid is placed for a set of time slots that is required by a certain event. Note that one event may generate multiple alternative sets of time slots. When an event requires two sequential time slots from t_{i0}, t_{i1}, and t_{i2}, for example, there are two possible alternatives, i.e., $\{t_{i0}, t_{i1}\}$ and $\{t_{i1}, t_{i2}\}$. A set of alternative sets of time slots is denoted by $AT_{ij} = \{AT^1_{ij}, AT^2_{ij}, \ldots, AT^k_{ij}\}$, where AT^k_{ij} denotes the *k-th* alternative for e_{ij}. Since agents can bid for all possible combinations in a combinatorial auction, detailed preferences concerning the time and date of an event can be represented. To represent the bid, possible combinations of items are enumerated and a valuation is allocated to each of them. A set of combinations of items, S_i, that agent r_i can bid for is as follows:

$$S_i = \bigcup_{j|e_{ij} \in E_i} S_{ij}$$

$$S_{ij} = \{S^1_{ij}, S^2_{ij}, \ldots, S^k_{ij}\}$$

$$S^k_{ij} = \left(\bigcup_{l|t_{il} \in AT^k_{ij}} T_{lR_j} \right) \cup \{d_{ij}\}$$

The event d_{ij} is a dummy item to express exclusivity among alternatives for identical events. S_{ij} is a set of alternatives for e_{ij}. A certain alternative S^k_{ij} consists of a dummy item and time slots for resources R_j which are restricted by S^k_{ij}. For instance, a certain event e_{ij} = ((*one slot from* $\{t_1, t_2\}$), $\{r_1, r_2\}$, 100) generates a set of alternatives $\{\{t_{11}, t_{21}, d_{ij}\}, \{t_{12}, t_{22}, d_{ij}\}\}$. "*one slot from* $\{t_1, t_2\}$" can be represented as $\{(t^0_1 \le T^0_{ij} \le t^0_2) \vee (t^0_2 \le T^1_{ij} \le t^1_2)\} \wedge (T^1_{ij} - T^0_{ij} = 1)$, where t^0_1 *and* t^1_1 (t^0_2 *and* t^1_2) denote the start time and the end time of a time slot t_1 (t_2), respectively. The allocation of the valuation enables various preferences to be represented. If an agent wants to hold an event e_{ij} earlier, it should set the difference in valuation between two bids, i.e., a bidding price for $\{t_{11}, t_{21}, d_{ij}\}$ is 110 and for $\{t_{12}, t_{22}, d_{ij}\}\}$ is 100.

The final schedule is a solution of the following formula:

$$\underset{\chi}{\arg\max} \sum_{j|S_{ij} \in \chi} v_{ij} \tag{1}$$

Here, χ is defined as follows:

$$\chi = \{S_{ij} \subseteq S_i \mid S_{ij} \cap S' = \emptyset \text{ for every } S_{ij}, S' \in \chi\} \tag{2}$$

The solution can be obtained by using various search techniques. In our work, we adopt the LDS (Limited Discrepancy Search) [4] and use the algorithm described in [10], which can quickly search for a high quality solution.

4 Introduction in Constraints to a Combinatorial Auction

To represent the number and order constraint based on the formalization described in Section 3, we must enumerate all possible combinations that take such constraints into consideration. As a result, the computation time increases exponentially since there is a dramatic increase in the number of combinations of items. Therefore, another way to express constraints, without having to enumerate combinations, is needed. In this section, we try to express a combinatorial auction as a Mixed Integer Programming (MIP) problem to avoid explosive increases in combinations of items.

4.1 Formalization of a Combinatorial Auction as an MIP Problem

First, we formalize a combinatorial auction as an MIP problem according to [1]. It is assumed that there are m items denoted by $M = \{g_1, g_2, ..., g_m\}$ and n bids denoted by $B = \{b_1, b_2, ..., b_n\}$. The bid is denoted by $b_i = < S_i, p_i >$, where S_i is a set of items and $S_i \subseteq M$; p_i is a bidding price for S_i and $p_i \geq 0$[1]. The winner determination problem in a combinatorial auction can be formalized as follows:

$$\max \sum_{i=1}^{n} p_i x_i \quad (x_i \in \{0,1\})$$
$$\sum_{i|g_j \in S_i} x_i \leq 1 \quad (g_j \in M)$$

If b_i wins, $x_i = 1$, if not, $x_i = 0$. Namely, to obtain the solution, agent should solve the above problem and regard b_i as a winning bid when $x_i = 1$.

[1] p_i is calculated from the valuation v_{ij} for S_i. In this paper, we do not discuss a calculation method of it.

4.2 Representation of Number Constraint

The number constraint is represented such that the number of winning bids must be n'_0 ($n'_0 \leq n_0$) or less for a set of n_0 bids $B_0 \subseteq B$. The definition is given as:

$$\sum_{i|b_i \in B_0} x_i \leq n'_0$$

The number constraint is useful for managing the exclusivity of bids that are generated from an identical event. The variation of bids for a certain event e is denoted as $B_e = \{b_{ei}|b_{ei}\ is\ a\ bid\ for\ event\ e,\ b_{ei} \in B\}$. The number constraint for exclusivity is represented as follows:

$$\sum_{i|b_{ei} \in B_e} x_i + x_{B_e} = 1 \tag{3}$$

Here, if all bids included in B_e do not win, $x_{Be} = 1$, and if not, $x_{Be} = 0$. Though it is possible to represent the constraint without x_{Be}, i.e., $\sum_{i|b_{ei} \in B_e} x_i \leq 1$, x_{Be} is required in order to represent the order constraint.

4.3 Representation of Order Constraint

The order constraint is one for describing the order of multiple events and the interval between them. In this section, an example case regarding the order and interval between two events, e_1 and e_2, is given. It is assumed that $e_1 <_l e_2$; that is, the end time of e_1 is l slots ahead of the start time of e_2. B_{e1} and B_{e2} are a set of bids for events e_1 and e_2, respectively. Let $t^0_{b_i}$ be the start time of event e required by b_i and $t^1_{b_i}$ be its end time required by b_i.

$$f = -\sum_{i|b_i \in B_{e_1}} t^1_{b_i} x_i + \sum_{i|b_i \in B_{e_2}} t^0_{b_i} x_i \geq l \tag{4}$$

If either bid for both events does not win, equation (4) will not be satisfied. For example, when a bid for e_1 wins the item and a bid for e_2 does not win,

$$-\sum_{i|b_i \in B_{e_1}} t^1_{b_i} x_i < 0 \quad and \quad \sum_{i|b_i \in B_{e_2}} t^0_{b_i} x_i = 0$$

Accordingly, equation (4) is not satisfied. To solve this problem, we use equation (5):

$$f + (\max\{t^1_{b_i}\ |b_i \in B_{e_1}\} + l)x_{B_{e_2}} \geq l \tag{5}$$

Even if $\sum_{i|b_i \in B_{e_2}} t^0_{b_{ij}} x_j = 0$, equation (4) will be satisfied since x_{Be2} to equation (3).

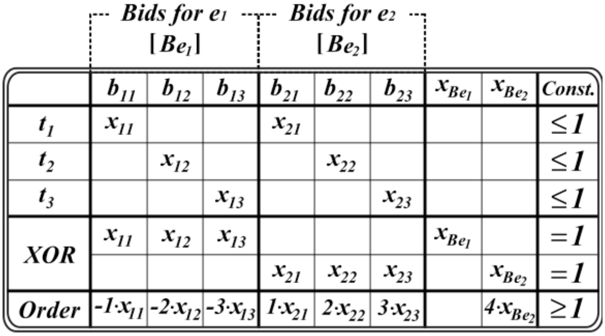
We can set the minimum interval l_{min} and maximum interval l_{max} by describing the interval between events as $l_{min} \leq f \leq l_{max}$. For instance, considering e_1 and e_2, which require one time slot from t_1, t_2, and t_3 and same participants, we represent the order constraint for these two events. Moreover, e_2 is $l_{min} = 1$ *slot*. In this case, the constraint is described as follows:

$$-x_{11} - 2x_{12} - 3x_{13} + 2x_{22} + 3x_{23} + 4x_{B_{e_2}} \geq 1$$
$$\sum_{i|b_i \in B_{e_1}}^{3} x_{1i} \leq 1, \sum_{i|b_i \in B_{e_2}}^{3} x_{2i} + x_{B_{e_2}} = 1$$
$$x_{11}, x_{12}, x_{13} : the \quad variables \quad for \quad e_1$$
$$x_{21}, x_{22}, x_{23} : the \quad variables \quad for \quad e_2$$

4.4 Formalization of a Combinatorial Auction with Constraints

A combinatorial auction can be formalized regarding some combinations of items and all constraints:

$$\max \sum_{i=1}^{n} p_i x_i \quad (x_i \in \{0,1\})$$
$$\sum_{i|t_j \in S_i} x_i \leq 1 \quad (t_j \in M = \{t_1, t_2, ..., t_m\})$$
$$c_{11}x_1 + ... + e_{1n}x_n + x_{n+1} = 1$$
$$\cdots\cdots$$
$$c_{d1}x_1 + ... + e_{dn}x_n + x_{n+d} = 1$$
$$t_{11}x_1 + ... + t_{1n}x_n + t_{1n+1}x_{n+1} \geq l_1$$
$$\cdots\cdots$$
$$t_{r1}x_1 + ... + t_{rn}x_n + t_{rn+1}x_{n+r} \geq l_r$$



	Bids for e1 [Be_1]			Bids for e2 [Be_2]					
	b_{11}	b_{12}	b_{13}	b_{21}	b_{22}	b_{23}	x_{Be_1}	x_{Be_2}	Const.
t_1	x_{11}			x_{21}					≤ 1
t_2		x_{12}			x_{22}				≤ 1
t_3			x_{13}			x_{23}			≤ 1
XOR	x_{11}	x_{12}	x_{13}				x_{Be_1}		$=1$
				x_{21}	x_{22}	x_{23}		x_{Be_2}	$=1$
Order	$-1 \cdot x_{11}$	$-2 \cdot x_{12}$	$-3 \cdot x_{13}$	$1 \cdot x_{21}$	$2 \cdot x_{22}$	$3 \cdot x_{23}$		$4 \cdot x_{Be_2}$	≥ 1

Fig. 1. A Scheduling Problem based on our Formalization

$$c_{ij} = \begin{cases} 1 : b_i \in B_{e_i} \\ 0 : otherwise \end{cases}$$

$$t_{ij} = \begin{cases} t_{b_i}^1 : b_i \ \ is \ \ a \ \ bid \ \ for \ \ a \ \ precedent \ \ event \\ t_{b_i}^0 : otherwise \end{cases}$$

According to the above formalization, constraints on the number of items, exclusivity of bids, and the order of events can be represented. The purpose is to maximize the sum of the valuations of successful bids.

Figure 1 shows simple example which represents a scheduling problem based on our formalization. In this example, we assume that there are events e_1 and e_2 ($e_1 <_1 e_2$) each of which require one time slot from t_1, t_2, t_3. The variation of bids for events e_1 and e_2 are denoted as $B_{e1} = \{b_{11}, b_{12}, b_{13}\}$ and $B_{e2} = \{b_{21}, b_{22}, b_{23}\}$, respectively. Each of the columns indicates variables with respect to a bid excepting the two columns on the right. The first three rows (t_1, t_2, t_3) indicate the status of each time slot. For instance, variable x_{11} is required by bid b_{11}. Each of these three rows is constrained according to the formalization, that is "≤ 1". The fourth and fifth rows (combined as row "*XOR*") are used to deal with the number constraints, more specially exclusivity. Here, the value of variables x_{Be2} and x_{Be1} is set according to equation (3), i.e., if $x_{11} + x_{12} + x_{13} = 1$, $x_{Be1} = 0$. Thus, even if all the bids for event e_1 cannot win, the fourth row can satisfy the constraint. The last row is used to deal with the order constraint. Here, according to equation (5), the coefficient of x_{Be2} is the sum of the maximum number of time slots "3" and interval "1". Given that b_{12} and b_{21} are accepted, the values of $x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{Be1}$, and x_{Be2} can be set to 0, 1, 0, 1, 0, 0, 0, 0, respectively. In this case, the order constraint is not satisfied, since $(-1\times0) + (-2\times1) + (-3\times0) + (1\times1) + (2\times0) + (3\times0) + (4\times0) \geq 1$. Thus, our formalization can represent that b_{12} and b_{21} are never accepted simultaneously. In the end, solving a scheduling problem comes down to finding the value of each variable which can satisfy all constraints, while the sum of valuations of successful bids must be maximized. To solve the problem represented in the form of a variable table Figure 1, we use the MIP solver offered by GLPK (GNU Linear Programming Kit) [6].

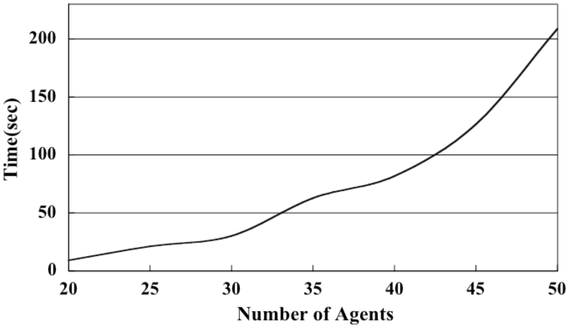


Fig. 2. Computation time over number of bids using LDS

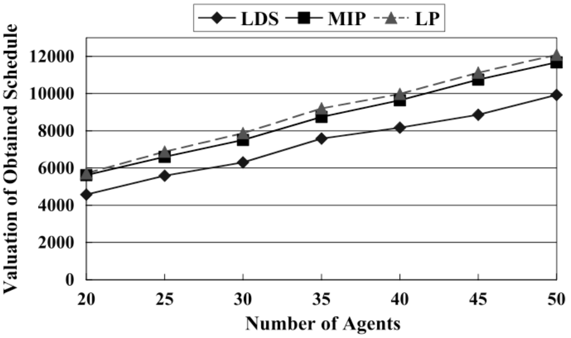


Fig. 3. Valuation over number of agents (MIP: 1800 sec.)

5 Evaluation

Our scheduling method was experimentally evaluated in terms of the computation time, the valuation, and the satisfaction rate. Satisfaction rate means the ratio of won events to all desired events. The number of time slots per agent was set to 40 (the sum of normal working hours in a week). The scheduling problem was generated randomly under uniform distribution. Though the number of participants and the length of every event were fixed, the start times were unfixed. For each number of agents, we generated 10 problem instances where we varied the number of agents from 2 to 50. The problem was solved in two ways; using the LDS method based on the formalization described in Section 3, and using the MIP solver based on the formalization described in Section 4. Because solving a problem by using an MIP solver takes a lot of time, the calculation was terminated 1800 seconds (30 minutes) after. The best solution at that time was used. The results of the experiments are shown in Figure 2, Figure 3, and Figure 4. All experiments were performed on a PowerMac G4 (PowerPC G4 1GHz, 512MB) with a program written in Java and C.

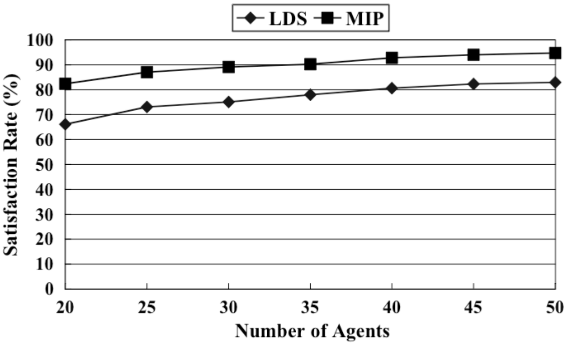


Fig. 4. Satisfaction rate over number of agents (MIP: 1800 sec.)

Figure 2 shows the computation time using LDS. This figure indicates that our method enables the largest problem to be solved in practical time. Although there are 50 agents in the largest problem, the computation time is about 210 seconds. This is sufficiently acceptable for use in practical scheduling systems.

Figure 3 shows the valuation using LDS, the MIP solver, and the LP (Linear Programming) solver which can obtain an optimal solution. In our experiments, the MIP solver can achieve better than 95% of the optimal solution through LP. Since the solution is calculated within a real number in LP, a unit of the schedule would be divided in the solution, e.g., "an event e_{ij} can be held during 2.5 time slots." Then, practically, it is impossible to use the solution obtained by LP. Generally, the valuation of possible optimal solutions is between the valuation of the MIP solution and LP solution. Therefore, we can consider that the MIP solution is almost optimal. Moreover, the LDS method can always achieve about 80% of the MIP solution. Figure 4 shows the satisfaction rate using the LDS method and MIP solver. As shown in this figure, the MIP solver can achieve a sufficient rate to use in practical scheduling. The LDS method can also achieve a high rate in all problem sets, but it is insufficient compared with the rate using the MIP solver. In Figure 4, as the number of agents increases, the satisfaction rate rises. This is because the ratio of private events to all events is increased due to the fixed number of shared events. The LDS method can find a solution with a small amount of effort compared with the MIP solver, but this solution is inadequate. However, the MIP solver performs well on the quality of the solution and the satisfaction rate. We think both of methods can be practically used. We should select the method based on the purpose. If agents desire to obtain a semi-optimal solution, they should use the MIP solver. Moreover, if agents need to express detailed preferences to obtain more satisfactory solutions for his/her user, the MIP solver is the appropriate way. However, if semi-optimal solutions are not necessarily desired and high computation costs are to be avoided, agents can solve the problem by LDS. But, we think that the waiting time for MIP solutions is not serious compared with the difficulty of solving the problem manually.

6 Related Work

The existing work on solving the meeting scheduling problem includes [12, 13]. In [12], each user's preferences are represented by values and weights for some attributes of each meeting. The scheduling is executed using a contract net protocol and a voting method. This method depends on heuristics; thus, whether it is possible to apply them to a large-scale problem is open to discussion. In [13], the scheduling problems are formalized as a Distributed Valued Constraint Satisfaction Problem. In this formalization, the user's preferences are represented by some constraints and weight. However, they use an ad hoc constraint relaxation method for an over-constrained problem. Therefore, the quality of solution cannot be guaranteed. In this paper, we improve the expressiveness of user's preferences by representing them as bids for them and constraints on a schedule. Moreover, for constraint relaxation, our method is based on an economical rationality; i.e., it is not ad hoc. Accordingly, we may possibly obtain a socially preferable schedule.

Hunsberger [7] uses a combinatorial auction protocol as a method for conflict resolution among agents. They use a combinatorial auction protocol for task planning. Namely, each agent places bids on possible tasks by considering the constraint on the date and time of each task. To determine the schedule, they use a search method that applies the algorithm described in [11]. In contrast, for determining the schedule, we use an MIP which enables us to consider several types of constraints.

7 Conclusion

In this paper, we formalized a scheduling problem considering many constraints as a combinatorial auction. Our contribution is that we represent every detailed constraint on the events by representing them for bidding within the framework of the Mixed Integer Programming (MIP) Problem. By solving the problem as a combinatorial auction, we were able to guarantee that the obtained schedule was the appropriate one and that it did not include impossible events. In our basic formalization, since many bids might be generated in a scheduling process, the computation time tends to be long. Therefore, we re-formalized a combinatorial auction as an MIP problem. As a result, the problem could be represented as a simple variable table. We could represent various constraints without creating a combinatorial explosion in the number of bids and obtain an appropriate schedule in practical time. We applied the LDS method and MIP solver to obtain a solution. We concluded that scheduling using the MIP solver based on our formalization is an efficient way of obtaining a semi-optimal schedule and solving the trade-off between the computation time to find a solution for a combinatorial auction and the expressiveness to represent a scheduling problem.

In this paper, we do not consider each agent's budget. Thus, to win desired event, a certain agent should bid at an expensively high price. One future direction of this study is how to allocate the budget to each agent at the initial state.

References

1. A. Anderson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner determination. *Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS2000)*, pages 39–46, 2000.
2. C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 527–534, 1999.
3. Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computation complexity of combinatorial auctions: Optimal and approximate approaches. *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.
4. W. D. Harvey and M. L. Ginsberg. Limited discrepancy search. *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 607–613, 1995.

5. H. Hattori, M. Yokoo, Y. Sakurai, and T. Shintani. A dynamic programming model for determining bidding strategies in sequential auctions: Quasi-linear utility and budget constraints. *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 211–218, 2001.
6. http://www.gnu.org/software/glpk/glpk.html.
7. L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. *Proc. of the 4th International Conference on Multi-Agent Systems (ICMAS2000)*, pages 151–158, 2000.
8. S. Rassenti, V. Smith, and R. Bulfin. Combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, Vol. 13, No. 2, pages 402–417, 1982.
9. Y. Sakurai and M. Yokoo. An average-case budget-non-negative double auction protocol. *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 104–111, 2002.
10. Y. Sakurai, M. Yokoo, and K. Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. Proc. of the *2nd ACM Conference on Electronic Commerce (EC-00)*, pages 30–37, 2000.
11. T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
12. S. Sen, T. Haynes, and N. Arora. Satisfying user preferences while negotiating meetings. *International Journal of Human-Computer Studies*, Vol. 47, pages 407–427, 1997.
13. T. Tsuruta and T. Shintani. Scheduling meetings using distributed valued constraint satisfaction algorithm. *Proc. of the 14th European Conference on Artificial Intelligence (ECAI2000)*, pages 383–387, 2000.

Multi-agent Based Home Network Management System with Extended Real-Time Tuple Space

Myung Jin Lee[1], Jun Ho Park[1], Soon Ju Kang[1], and Jeong Bae Lee[2]

[1] School of Electrical Engineering and Computer Science,
Kyungpook National University, Korea
{explr,zec}@palgong.knu.ac.kr, sjkang@ee.knu.ac.kr

[2] Dept. of Information and Computer, Sunmoon University, Korea
jblee@sunmoon.ac.kr

Abstract. A home network is a typical distributed and pervasive environment, which makes monitoring the status of all devices in the home and controlling such devices within a reliable time constraint both difficult and complex. Accordingly, for effective data communication with various distributed devices, the current paper presents the design architecture for a multi-agent-based home network management system with an extended tuple space concept, including real-time characteristics. In addition, a prototype is implemented based on the proposed architecture.

1 Introduction

A home network is a typical example of a fully distributed ubiquitous system and pervasive computing application that consists of various consumer devices and appliances with heterogeneous communication protocols and system softwares. As such, a home network management system needs to satisfy the following conditions. First, it should be able to remotely monitor the status of all devices in the home and effectively control devices within limited timing constraints. Second, it should be a reliable, regardless of the network load. Third, it should be a multi-agent (e.g. fire detection service agent, video stream recording agent, device management and control agent, GUI agent, etc.) based distributed system for controlling distributed devices effectively. Finally, it should be a user-friendly system for all types of users in the home. Thus, building a home network management system is both difficult and complex.

Based on the current home network industries, several communication protocols and middlewares have already been introduced and embedded into home appliances. As regards protocols, the IEEE1394 protocol is used for home theatre services[4], LonTalk or CAN(Controller Area Network) is used for home automation and device control[5, 6], and Bluetooth or irDA is used to support wireless mobile devices in the home or office. Meanwhile, for middlewares, HAVi, Jini, or VESA is used in home networks to cope with a distributed network environment.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 188-198, 2004.

However, none of these middlewares can meet the problems involved in implementing a network management system in the home. For example, HAVi and VESA are only dedicated to IEEE1394 devices and have no global repository to store the status information on all the devices in the home. Though Jini has an advanced repository scheme, its application network can become unreliable depending on the traffic, as it has a lookup server centralized repository architecture, plus it only supports web- based protocols, such as TCP/IP, HTTP, and FTP and has no idea about the variety of protocols used in a home network.

Accordingly, a tuple space[1] has been suggested as a solution to these problems, since it is a suitable network middleware for ubiquitous computing and includes spaces that represent a loosely coupled communication model for distributed application programming, where each process can communicate without considering the others. Thus, because a tuple space is appropriate for any application that has distribution or repository requirements[11], it can be applied to home networks[2]. Tuple space solutions, such as T Space[7] and JavaSpaces[8] that are extensions of the basic Linda Tuplespace framework, have been suggested as appropriate for real distributed networks. Yet, the internal message commands of a tuple space are too simple to realize reliable data communications between controlling devices and controlled devices in a home network. In addition, there is no consideration of priority and periodic or sporadic tasks in real-time constraints.

Therefore, the current study suggests an extended tuple space to solve these problems, and explains a multi-agent based home network management system using the proposed extended tuple space. First, the basic home network architecture is introduced in Section 2, then Section 3 discusses the concept of an extended tuple space for specific use in a home network environment. The design architecture of the proposed multi-agent-based home network management system using the extended operations in a tuple space is described in Section 4, along with a prototype implementation based on the proposed concept. Finally, some conclusions are given in Section 5.

2 Replicated Tuple Space-Based Home Network Architecture

Fig. 1 shows the proposed replicated tuple space model in a home network, as mentioned in another paper[2]. In this figure, the home network consists of several room subnets(sub-networks), where each room server has multiple control applications and agents that manage real devices and a tuple space that is shared by several agents and control applications. All the tuple spaces in the home network are interconnected to each other through an event-messaging model. In this architecture, any changes in the device status and control commands in one room server are transferred to all the other servers, that is, the data for each tuple space is replicated in all the other spaces. As such, an agent can autonomously manage a specific device using the distributed tuple spaces and the status of all devices can be transparently accessed, regardless of the agent's location. Because of this location transparency, a newly attached device or agent can easily obtain information on another agent by either reading or taking information from the local tuple space. As a result, all agents are loosely coupled to each other. Therefore, this architecture can provide a more reliable and intelligent service based on easy access to the information in the local tuple space.

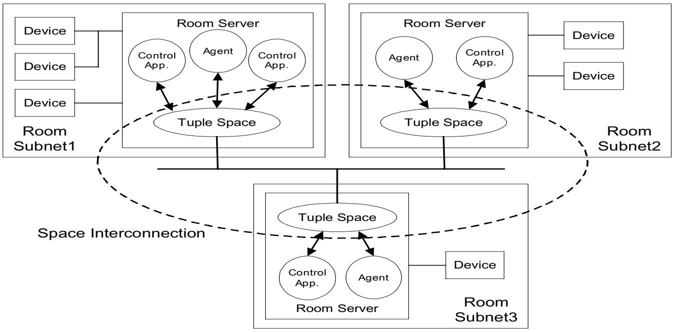


Fig. 1. Replicated tuple space in home network

3 Extended Real-Time Tuple Space Architecture

3.1 Requirements of Tuple Space in Home Network

To realize the proposed home network architecture mentioned in the previous section, certain extended operations need to be newly defined on top of the basic operations in the current tuple space concept.

First, a home network includes various consumer devices and services that each have a dead line and transmission priority level. For example, in a fire detection system, the events and commands of an alarm device must have a higher priority than those of other devices. In this case, the tuple space should store tuples along with their priority, and transfer tuples according to their priority to the control application within a dead line.

Second, the tuple space needs the support of an intelligent update mechanism for status information to reduce the network load. The data flows in a home network are classified into control commands, event, and status information, plus status information is divided into periodic and sporadic data. To reduce the network load, an extended update operation is needed that only updates the periodic or sporadic status information when any meaningful change occurs. Moreover, in the case of periodic status information, update-time adjusting is also useful to reduce the network load.

Third, the tuple space must support a transaction mechanism guaranteeing that the control sequence is operated at once. Most device control commands in a home network make a request and wait for a response. In this case, the command provokes “take” and “write” operations successively, while the response also creates “take” and “write” operations in order. In the middle of this transaction, if other devices make a request to the device involved in the original transaction, the control sequence can become disordered, making the systems unreliable. Therefore, in a home network, distributed group transactions are required.

3.2 Overview of Proposed Architecture

The current study suggests new architecture and extended operations for a tuple space to realize the requirements mentioned above. Fig. 2 shows the proposed extended tuple space architecture for a home network that consists of a priority-based event channel, space manager, network manager, and tuple space.

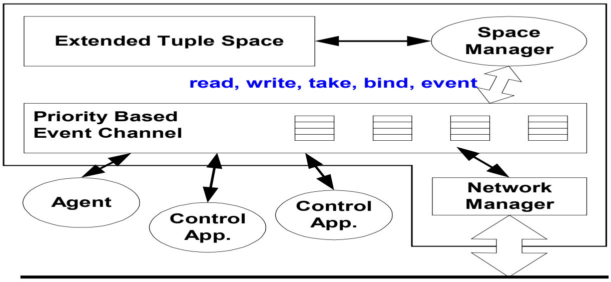


Fig. 2. Proposed extended tuple space architecture with real-time characteristic

The tuple space acts as temporal database that contains the control commands and status information generated by agents and control applications in the form of a tuple. The space manager is the interface for the tuple space and supports both the basic operations, such as read, write, and take and the extended operations for control transactions. In addition, the space manager plays a part in event managing by transmitting an event to the software elements in the control server. The priority-based event channel[3] then routes all requests, responses, and events to the software components of the control server, such as the space manager, agents, control applications, and network manager. The event channel schedules the requests based on their priority using a priority queue and prioritized threads to guarantee real-time requirements. Meanwhile, the network manager controls the network connections and serializes or de-serializes the tuple object to configure distributed networks. As such, in this architecture, all communication data and control commands between the software components are registered in a tuple space and scheduled by the priority-based event channel.

3.3 Real-Time and Load Reduction Data Updating Operation

To support the priority-based event scheduling and intelligent update mechanism for periodic or sporadic status information, an extended *"write"* tuple space operation is proposed.

```
write(tuple("data_class","device_name","device_position
","status",… ), priority, update_time,
min_offset_value, max_offset_value);
```

The *"data_class"* field in a tuple parameter indicates the data classification, such as control command and status data, while the *"priority"* parameter represents the importance of the device management when being scheduled by the priority-based event channel. Thus, based on the "*priority*" parameter, each event can be separated and processed according to its priority. The *"update_time"* parameter, which is managed by the space manager, indicates the minimum update period to maintain system reliability. If an agent reads a tuple within the *"update_time"* period, the space gives a cached tuple. If the *"update_time"* has expired, the space manager sends a refresh event to the control application and a new tuple is recorded in the tuple space. The *"min_offset_value"* and *"max_offset_value"* parameters indicate the minimum and maximum offset value, respectively, for updating the tuple space. As such, these parameters help the space manager to discard data that does not exist within the offset boundaries, thereby preventing frequent updating of the *"update_time"* parameters with unnecessary data.

3.4 Reliable Data Transaction Management Operation

As mentioned in Sect. 3.1, since the tuple space needs to support a transition mechanism guaranteeing the control sequence, a new tuple space operation is proposed called *"bind"*.

```
bind(tuple1, tuple2, tuple3, priority);
```

The tuple1, tuple2, and tuple3 parameters represent tuples related to a specific transaction, where tuple1 is the parameter for taking, tuple2 is the parameter for writing, and tuple3 is the parameter representing the result of a binding transaction. Then, the priority parameter represents the binding priority

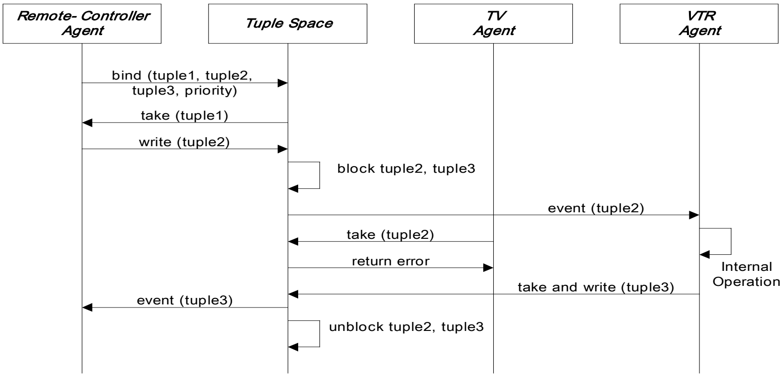


Fig. 3 Sequence example of binding transaction

In this operation, two applications take part in the *"bind"* operation and use two tuples. When a controlling (source) application creates a *"bind"* operation, the space manager throws the event to the controlled (target) application. When the controlled application receives the *"bind"* operation from the space manager, it performs the request and writes the result of the request in the tuple space. The space manager then returns OK to the first control application. The space manager protects the tuples involved in the *"bind"* operation by checking their priority. If other agents or applications create a *"take"* or *"write"* operation in the tuple, the space manager returns a null. Yet, in the case of a *"read"* operation, the space manager returns the tuple. However, during a *"bind"* operation, the space manager blocks the application, then when every process has been completed, it returns OK. If a higher priority agent or application interrupts an ongoing *"bind"* operation, the space manager returns an error to the current application and the *"bind"* operation is canceled.

Fig. 3 shows a sequence example of a binding transaction, where a remote-controller sends a command to the VTR agent and receives a response. As such, the remote-controller agent creates a *"bind"* operation in the tuple space. This figure also shows that a lower priority operation is disregarded in the tuple space, and the space returns an error. However, if the TV agent's tuple command had had a higher priority than the *"bind"* operation, the tuple space would have returned an error to the remote-controller agent and cancelled the bind relation, then the tuple space would have performed the operation with the higher priority, although not shown in Fig.3.

4 Case Study: Multi-agent-Based Home Network Management System

4.1 Home Network Prototype Configuration

The proposed extended real-time tuple space architecture was implemented in a home network, as shown in Fig. 4. Three subnets were organized as the real testbed of a home network, and each subnet was connected to an IEEE1394-based backbone network through individual room servers. Two tiny diskless SBCs(Single Board Computers) based on an X86 CPU were used as the first and second room servers, and one PC with a TI IEEE1394 adaptor board was used as the third room server.

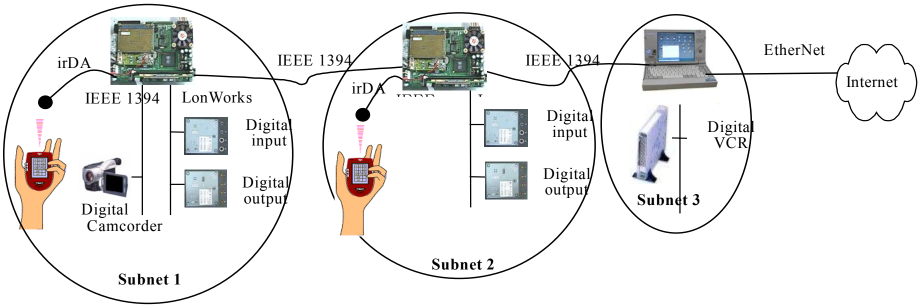


Fig. 4. Hardware configuration of home network prototype

The core components in subnet 1 and 2 were implemented under a Linux operating system with JDK. For the backbone and subnet, MotionIO OHCI-104 IEEE1394 adaptors were installed into the room servers[2] for subnet 1 and 2. In addition, a DI-10(for digital input devices) and DO-10(for digital output devices) were connected into each room server as LonTalk-based control network nodes. Gesytec's LonTalk adaptor boards were used to accommodate a LonTalk network in the subnet. Two digital multimedia devices (Sony DCR-TRV510 digital camcorder and Sarotech FHD-353 80GB HardBox) were used as the IEEE1394-based home devices and connected to each subnet. The room server in subnet 3 used Windows OS with jdk1.3, which was connected to the Internet through an Ethernet adaptor. To support the proposed extended tuple space, the T Space implemented by IBM was used, while the space manager was implemented as a wrapper of T Space. For example, the *"bind"* operation mentioned above was implemented using two swap operations of T Space. However, all the device drivers for each protocol adaptor were implemented in C and connected to Java using a JNI(Java Native Interface).

Based on the hardware configuration shown in Fig. 4, the proposed extended real-time tuple space architecture was tested by implementing home network management agents. Two kinds of agent program were used to implement user-friendly management under the proposed tuple space. The first agent was for editing and archiving the GUI(Graphical User Interface) icons of new consumer devices into the home network. This GUI editor agent is used by appliance manufacturers or home network suppliers to maintain their products remotely. The other agent was for remote controlling and monitoring the consumer devices in the home network using the GUI icons built by the GUI editor agent. These agents cooperated with each other through the tuple spaces. Next, the embodiment of the tuple space and operational architecture of the two agents are explained.

This multi-agent-based home network management is a typical application of the proposed tuple space, as shown in Fig 1. All agents can pursue their tasks without considering other agents or subnets by communicating with the repository in the subnet to which they belong. The synchronization and coordination between the agents or other subnets are then asynchronously solved by the distributed tuple spaces.

4.2 Embodiment of Space Manager with Real-Time Tuple Space

As the space manager duplicates the state information and control functions into distributed resource repository spaces, a distributed data structure can be easily designed to reflect the local structure.

Fig. 5 shows a coarse granulated example of a classification hierarchy of consumer devices in a home network. There are a lot of common control variables and status variables in consumer devices. The following is an example of device status information in a tuple space according to the modeling information:

("Light",Room2,5 ,"On",6,050401/113442,3,8,Lon,GE,"PowerOn-ON")
("TV",Room2,2,"On",6,050401/113442,3,8,iLink,Samsung,","PowerOn-OFF,Channel,...")
("AirCon",Room2,1,"On",6,050401/113442,3,8,Lon,LG"PowerOn-Off, TempLevel,...")
("Camcorder","Room1",5,"On",6,050401/113442,4,5,iLink,Sony,"RW,PLAY,FF, STOP,...")

The information in a tuple space is highly abstracted and coarse information about each device. Since all the control functions for a device are defined in its subnet, if an agent wants to control a particular device in a home network, it has to look-up the control variables for the target device in the device tuple in the repository of the target subnet, then the agent can invoke the control functions using a remote method invocation through the help of the device proxies in the target subnet.

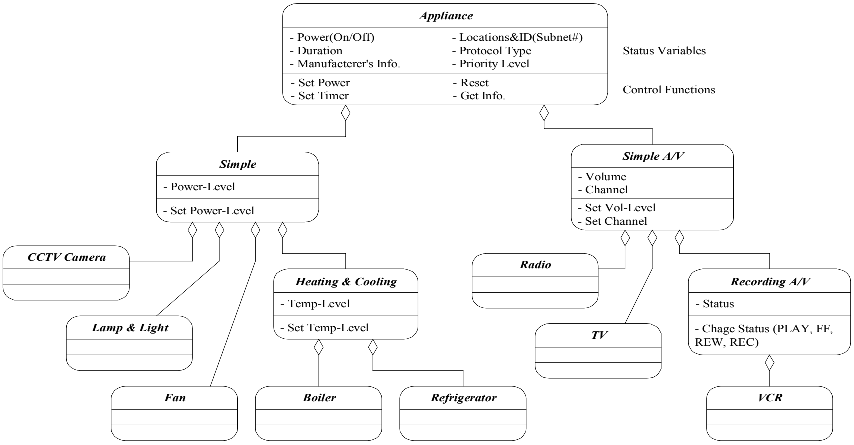


Fig. 5. Object-oriented hierarchy of consumer devices in home network

Each space manager shown in Fig. 2 supports the extended real-time tuple-based operations of retrieving and archiving device registry information for application agents in the subnet or device proxies inside a subnet. For example, a light management application agent can control the light status as follows: *write(tuple("Light", Room2, 5, "Off", ...), priority, update_time, min_offset_value, max_offset_value)*. Consequently, this distributed real-time tuple space model connects all physical and virtual device proxies and provides transparent access to all devices in the home network, regardless of the physical location of a device.

4.3 Graphical User Interface Editor Agent

The GUI agent is an agent that runs under the proposed real-time tuple space and builds GUIs for newly defined devices in the network. This agent has several functions for building a GUI, including an icon editor, GUI composer, and common libraries for status variables and control functions.

Fig. 6 illustrates the user interface of a working GUI agent. Some libraries include commonly used status variables and control variables according to the object-oriented modeling result of device information. Using the elements in each library, the user can pick a status variable or control function icon, then drop it into a newly designed device. For example, a VTR GUI consists of 5 control functions - forward, rewind, record, stop, play, and one status variable - power(on/off). Plus, the user can also design a new status variable icon, control function icon, and RMI(Remote method invocation) program using the icon editor.

To deposit the designed GUI into the home network, the user presses the b(broadcast) button on the far right. Then, the agent converts all the GUI information into a tuple-based representation and broadcasts it to all tuple space managers in the distributed tuple space. In the case of the control functions, the agent sends the control codes to the target subnet, then registers them in the device pool of that subnet.

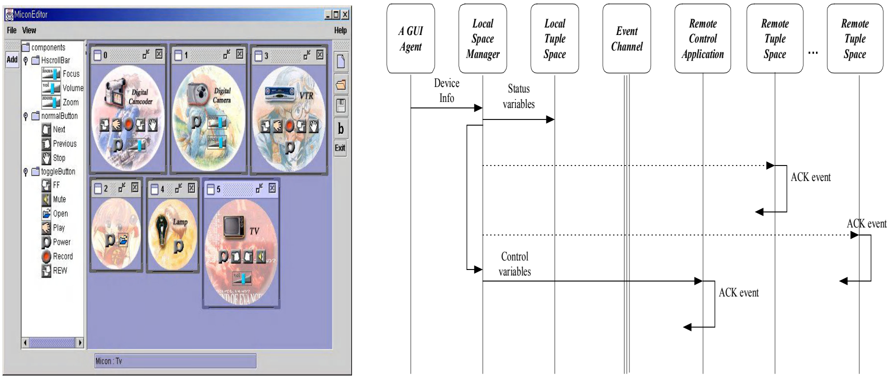


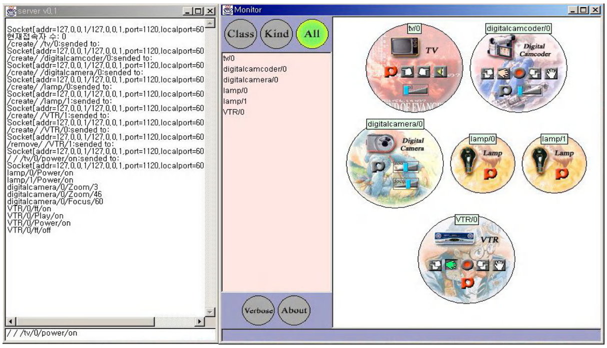
Fig. 6. User interface and event sequence diagram of GUI editor agent

As shown in the sequence diagram in Fig 6, the agent archives the newly designed device information, including the GUI icons, in the local tuple space through the space manager. At this time, the space manager broadcasts the device information to all tuple space managers in the home network. Then, the control variables for the newly defined device are sent and registered in the device pool of the subnet to which the controlled device belongs.

4.4 Home Network Management Agent

Fig. 7 shows a Java Beans-based user interface for a home network management agent. Users can invoke this agent using a display device, such as a TV or PC without considering the location of the display device or other agents running on the home network.

The icons in the right-hand window can be built using the GUI editor agent mentioned above or by the manufacturers of the device according to standard guidelines defined in the proposed middleware. Using the icons, the user can easily control a device by pressing the sub icons within the large circled icons. For example, to turn off lamp 1 in a subnet(room), press the “**Power(P)**” sub icon within the icon “lamp/1”. The left-hand window shows some status information for the home network, such as the current number of agents, plus a list of active devices and their status, as simple text. The middle window shows the list of devices in a particular subnet and the icons for handling group-based commands and graphical icons for the devices.



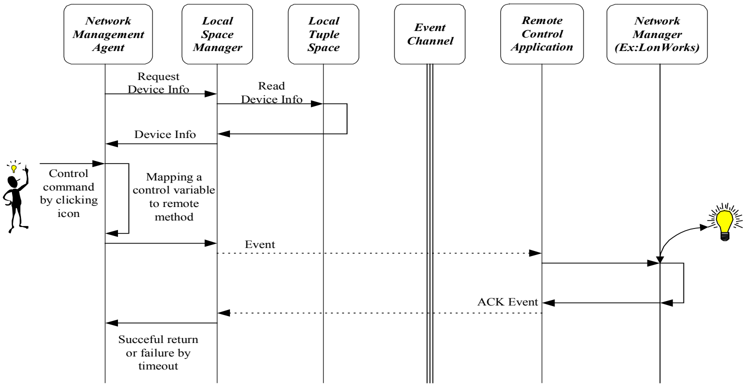
```
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
현재접속자 수: 0
/create/ /tv/0:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /digitalcamcoder/0:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /digitalcamera/0:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /lamp/0:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /lamp/1:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /VTR/1:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/create/ /VTR/0:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/remove/ /VTR/1:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
/ / /tv/0/power/on:sended to:
Socket[addr=127.0.0.1/127.0.0.1,port=1120,localport=60
lamp/0/Power/on
lamp/1/Power/on
digitalcamera/0/Zoom/3
digitalcamera/0/Zoom/46
digitalcamera/0/Focus/60
VTR/0/ff/on
VTR/0/Play/on
VTR/0/Power/on
VTR/0/ff/off
```

```
/ / /tv/0/power/on
```

```
tv/0
digitalcamcoder/0
digitalcamera/0
lamp/0
lamp/1
VTR/0
```

Fig. 7. User interface of home network management agent

Fig. 8 shows the event sequences after the activity of a home network management agent. First, the agent looks-up (using a read operation in a tuple space) all the re source information from its local tuple space and repeats the look-up activity item-by-item according to the device constraints. When the user controls a device using the GUI, the agent invokes a remote method mapped to the selected control variable.



Network Management Agent
Local Space Manager
Local Tuple Space
Event Channel
Remote Control Application
Network Manager (Ex:LonWorks)
Request Device Info
Read Device Info
Device Info
Control command by clicking icon
Mapping a control variable to remote method
Event
ACK Event
Succeful return or failure by timeout

Fig. 8. Event sequence describing activity of home network management agent

5 Conclusion

The current paper proposed the extended tuple space architecture and operations to support the real-time characteristics of a home network. The proposed update management in a tuple space using an extended write operation can reduce the network load, while the proposed bind operation can support reliable actions in the control processes and guarantee the transaction sequence. The proposed architecture was implemented and tested in a home network prototype environment. According to the case study implemented for a multi-agent-based home network management system, a user-friendly and convenient multi-agent application development was demonstrated under the proposed extended real-time tuple space architecture. Based on this architecture, the use of other control applications, such as fuzzy or feedback control, will also be investigated in a home network,

Acknowledgements. This research was supported by University IT Research Center Project.

References

1. N. Carriero and D. Gelernter, : Linda in Context, Comm. ACM, vol.32, no.4 (Apr. 1989)
2. Soon-Ju Kang, Jun-Ho Park and Sung-Ho Park, : ROOM-BRIDGE: A Vertically Configurable Network Architecture and Real-Time Middleware for Interoperability between Ubiquitous Consumer Devices in Home, Lecture Notes in Computer Science 2218, (Jan. 2001) 232-251
3. Jae Chul Moon, Jun Ho Park, and Soon Ju Kang, : An Event Channel-Based Embedded Software Architecture for Developing Telemetric and Teleoperation Systems on the WWW, Proc. IEEE Real-Time Technology and Applications Symp., Jun. Vanquaver, Canada (1999)
4. IEEE std 1394-1995, Standard for a High Performance IEEE 1394.0
5. http://www.lonmark.org
6. ISO 11898:1993 : Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication, http://www.iso.org
7. P. Wyckoff, S. McLaughry, T. Lehman, D. Ford, : T Spaces, IBM Systems Journal, Volume 37, Number 3 (Aug. 1998) 454-474
8. JavaSpace Specification, Revision 0.4, http://java.sun.com/products/javaspaces/specs/js.ps, Sun Microsystems, Inc.
9. P. Ciancarini, A. Knoche, R. Tolksdorf, and F. Vitali, : PageSpace: an Architecture to Coordinate Distributed Applications on the Web, Computer Networks and ISDN Systems 28, No. 7-11 (May 1996)
10. P. Ciancarini, A. Knoche, D. Rossi, R. Tolksdorf, and F. Vitali, : Coordinating Java Agents for Financial Applications on the WWW, Proceedings of the 2nd Conference on Practical Applications of Intelligent Agents and MultiAgent Technology (PAAM) (April 1997) 179-193
11. J. Lehman, Alex Cozzi, Yuhong Xiong, : Hitting the distributed computing sweet spot with TSpaces, Computer Networks 35 (2001) 457-472

Modelling Multi-aspect Negotiations in Multiagent Systems Using Petri Nets

Mateusz Lenar and Aleksander Zgrzywa

Wroclaw University of Technology, Department of Information Systems,
Wybrzeze Wyspianskiego 27
50-370 Wroclaw, Poland
{lenar, zgrzywa)@pwr.wroc.pl

Abstract. Agent-based technologies are used to develop concurrent systems running in open environment. Research in multiagent systems is concerned with modelling and designing of communication between autonomous agents to make it successful. This work applies to negotiation as a form of communication between agents and concurrency of this process. We propose multi-aspect negotiation, which may be run concurrently between agents in autonomous bilateral negotiation threads using different negotiation strategies. High level Petri nets are known model to design concurrent and distributed systems working in the open environment such as the Internet. We have presented how to use the Petri nets formalism to design negotiating system. On the basis of implemented real-world application we have carried out an experiment, which shows the relationship between chosen strategies and negotiation effectiveness and verbosity.

1 Introduction

When designing multiagent systems we have to pay attention to automatic conflict resolving. This is one of the most important fields in the area of Distributed Artificial Intelligence. It is used both in simple and complex problem solving. Each autonomous entity of the system bases on its inner knowledge and data in decision taking. It only sometimes is able to use directly other system knowledge. An access to other agents' skills and knowledge is available in information exchange or negotiation process. These processes may take place only when an agent can formulate its needs that are understandable to the other agent. Another problem arises here: how to establish an acceptable price to both sides of negotiated subject? It is necessary to design the communication language, which supports information transfer and proper identification of information content. To provide communication we use point to point negotiations, in which agreements may be reached relatively easily, due to two entities attendance only. This kind of negotiation can be run concurrently by every entity in multiagent system. The coordination between negotiation threads is controlled in two ways: the agent itself controls negotiation state and system state is controlled by

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 199-208, 2004.

system rules. This research is concerned with finding out what is the relationship between negotiation strategy, number of negotiated aspects and the result of negotiation. We have examined several strategies and tactics and analysed their influence on negotiation success.

2 Negotiation in Multiagent Systems

Multiagent system may be defined as a common platform where autonomous part of software can communicate, co-operate and reach their particular aims. In this paper we concentrate on co-operation between agents without going into details about system formal representation. We base on definitions of the multiagent systems and software agents presented in [3].

In agent-based systems, where autonomous agent compete with other agents, there often conflict situations arise. It is caused by autonomy and distribution of interacting agents, lack of centralized control and limited or lack of resources. Each autonomous agent determines its own goal, which is often in conflict with goals of other agents [1]. The designers of agent systems should predict such situations and prepare some mechanism, which could manage to resolve conflicts. The simplest way to do this is to implant negotiation mechanism into a communication protocol. Negotiation is defined as a process, "*by which group of agents communicate with one another to try and come to a mutually acceptable agreement on some matter*" [9]. Negotiation mechanism used in agent systems usually consists of protocols, strategies and deals. While constructing this mechanism we should take under consideration the following criteria: adequacy, efficiency, simplicity and agent's motivation to negotiate [10].

2.1 Negotiation Mechanism

Negotiation mechanism should be designed and judged by the following criteria [13]:

- Adequacy - have to be provided both in quantitative and qualitative sense. Negotiation mechanism must be constructed in complete way, giving all agents possibility to act in several ways. Every agent should be treated equally. Designed mechanism has to be rational, looking both from agent and system perspective.
- Efficiency - negotiation mechanism has to be Pareto optimal, that is there are no other result, that increases the value of negotiated matter to one agent and at the same time leaving the value for other agents at least at the same level,
- Self-motivation – some motivating functions should be provided to increase agents' activity to co-operate,
- Simplicity – communication language and negotiation mechanism have to be designed in simple way, should be easy to implement and easy to understand.

2.2 Negotiation Protocol

Negotiation protocol describes principles and rules of interaction between agents. To design the protocol we use existing low-level communication protocols and focus on definition of communicates semantics and we specify:

- number and types of participants,
- negotiation states and conditions of change,
- making offers sequence,
- types of possible deals.

2.3 Negotiation Strategy

Negotiation strategy is a set of rules and functions that is used in negotiation process. The strategy covers aims and manners to reach the mutually acceptable state of the system. In this paper we focus on two strategy types:

- time-dependent – the agreement has to be reached before specified moment in time or in specified numbers of iterations;
- resource-dependent – an agent has an access to limited resources or skills that is why it does not weaken its demands below specified level.

There also exist other types of strategies e.g. imitating strategies, where an agent wants to mislead other agents to maximize their own gain by minimizing gains of other agents.

In [10] we presented the formal model of negotiation, which was worked out on the basis of [2]. Some features of this model we present below. There exist a finite set of negotiating agents *S={A, B, .., N}* and a finite set of aspects being under negotiation (e.g. price, colour, delivery time). For the simplification, the aspects are numbered from 1 to m. Each aspect may have continuous values (e.g. delivery time) or discreet (e.g. colour names). During the negotiation process, the agents evaluate scoring function value for the contract and decide what to do next. When the value is acceptable, the agent tries to finalize the contract. Otherwise, the agent generates the counter offer. The scoring function has two arguments: a possible value of the aspect *j* and the moment of time; V_j^i : $[min_j, max_j], t \rightarrow [0, 1]$. Different functions may be used for each aspect. Basing on agent profile we have to determine relative importance of aspect *j* to agent *i* (w_j^i). These weights are normalized according to formula: $\sum_j w_j^i = 1$. Scoring function for the whole contract may be defined as $V^i = \sum_j w_j^i \cdot V_j^i(x_j)$, where x_j is a value of an aspect *j*

We have analysed three basic types of scoring functions. First function's type (1) is linearly weakened with time. This function is used in simple linear strategies. Second scoring function's type (2) is used in time dependent strategies. Using this strategy, an agent is softening its position faster when the time of negotiation is close to be run out. The third type of scoring function (3) is used in resource dependent strategies. It is used, when an agent wants to reach an agreement but it is limited with owned resources. The agent is weakening its position to the minimum value and never exceeds it. Basing on these functions we can model different strategies depend on what we want to achieve. In the real-world application pure strategies are very rare that is why mixed strategies are used.

In all equations listed below c_i is a constant ($i \in \{1, 2, 3, 4\}$).

$$f(x) = c_1 x + c_2 \tag{1}$$

$$f(x) = c_1 (x - c_2)^{c_3} + c_4 \tag{2}$$

$$f(x) = c_1 e^{c_2 (x - c_3)} + c_4 \tag{3}$$

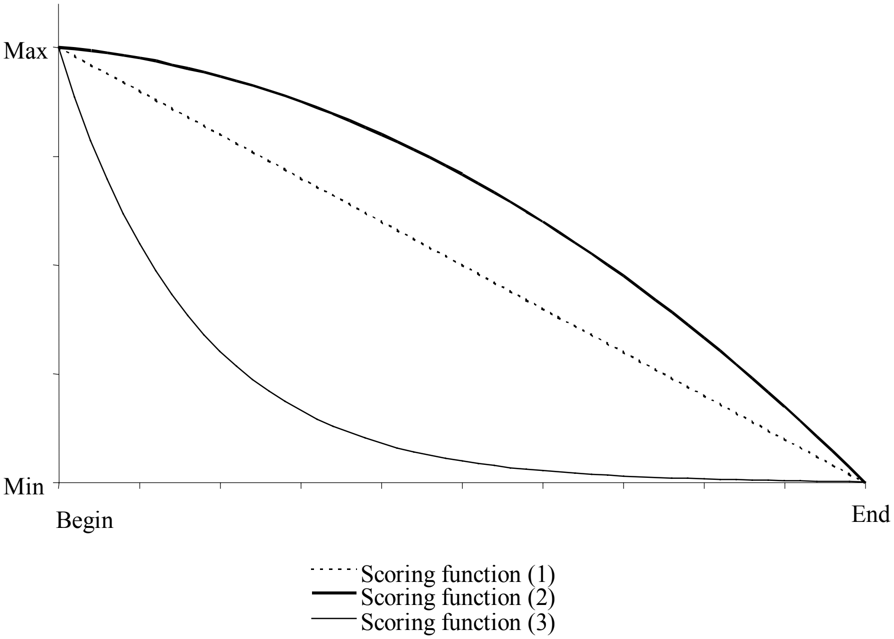


Fig. 1. Types of analysed scoring functions. One can observe different pace of decreasing function's value.

2.4 Negotiation Deals

The deal is defined as a result of the negotiation process. Below we describe basic types of negotiation deals [13], [14]. "*Pure deals*" are reached when every aspect of the deal is approved by both agents. “*Mixed deals*” are similar to “*pure deals*”, but in mixed deals execution plan is not specified. “*All-or-nothing deals*” are the special case of the mixed deals. There exist a job to be done and this job is indivisible and could be performed only by one agent. This deal brings profits to both sides but hires only one party to do the job. Another type of deals is “*semi-cooperative deals*”. Final agreement in these deals is not possible to achieve but the agents decide to cooperate as long as possible. This kind of deals brings only partial accomplishment of the aim to both agents. And finally, “*conflict deals*” are a special kind of deals when no agreement is reached.

3 Petri Nets

In recent years we have noticed constant development of Petri nets theory. To overcome some drawbacks of classic Petri nets high level Petri nets (HLPN) have been developed. There are many different kinds of these nets (e.g. coloured, timed, object, stochastic) and each of them is applied to a certain kind of system. HLPN’s originate from the idea of simple Petri nets and are expanded of some set of new features. These features make HLPN very attractive tool to model and simulate complex systems. Several frameworks and methodologies using Petri nets have been developed to model and simulate multiagent systems [4], [5], [7], [12], [16]. HLPN may be defined as:

$\mathrm{HLPN} = (\mathrm{P}, \mathrm{T}, \mathrm{A}, \mathrm{I}, \mathrm{C}, \mathrm{W_I}, \mathrm{M_0}, \mathrm{X}, \mathrm{O}, \mathrm{G}, \mathrm{F})$

P – places

T – transitions

$\mathrm{A} \subseteq (\mathrm{P} \times \mathrm{T}) \cup (\mathrm{T} \times \mathrm{P})$ – arcs

$\mathrm{I} \subseteq (\mathrm{P} \times \mathrm{T})$ – inhibitor arcs

$\mathrm{C} : \mathrm{P} \rightarrow \aleph$; capacity function;

$\mathrm{W_I} : \mathrm{I} \rightarrow \aleph$ – weight function

$\mathrm{M_0} : \mathrm{P} \rightarrow \mathrm{O}^*$ – initial marking function

$\mathrm{X} : \mathrm{A} \rightarrow \wp(\mathrm{V}) \setminus \varnothing$ – arc inscriptions; V – set of variable names

O – set of objects (colours)

G – guard function (input bindings of transition)

F – binding function (output bindings of transition)

HLPN’s are very appropriate method to model true concurrency in distributed systems. Another advantage of HLPNs is the ability to model multiagent systems on different stages of system development: from the simplest model where an agent is associated with the transition that occurs in the net to the hierarchical model that consists of autonomous nets.

HLPNs are advantageous because:

- can be used as a specification of the system or a presentation;
- have the ability to describe simultaneously states and actions using very few primitives;
- integrate the description of data manipulation with the description of synchronisation and control;
- represent true concurrency;
- support abstraction and refinement (object, hierarchical),
- provide multiple use of shared code.

4 Modelling Negotiations between Agents Using Petri Nets

As an example we want to present negotiation between two agents modelled by place transition nets (P/T nets).

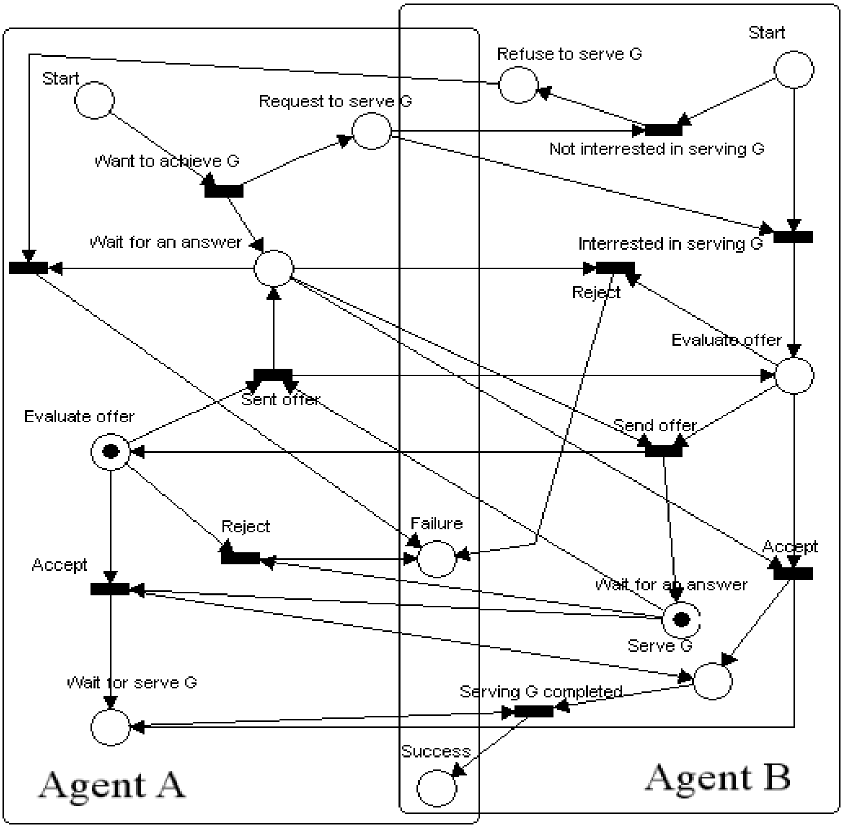


Fig. 3. The state of the system after a transition "Send offer" fired by Agent B. Agent A evaluates the offer and agent B waits for counteroffer, acceptance or rejection.

Let us consider two agents: A – which wants to buy a car and B – which want to sell a car. These two agents exchange offers with each other. Agent A has a goal G and wants from agent B to serve G (send a specific car). This process is run until approval or rejection of the offer. The pace of negotiation depends on negotiation mechanism, especially on agent's strategy. This simple example shows the practical use of Petri nets and formal model of negotiation to design and describe negotiation process. This abstract model can be defined in more detailed way. Each transition may be described as a subnet. So we can design complex models using simple and easy to understand graphical representation. The presented formal model of negotiation allows us to define the set of negotiation aspects, scoring functions and importance weights.

5 Multi-aspect, Bilateral Negotiation in Car-Seller Application

5.1 System Description

In this section we present an implemented multiagent system. Our system is placed in car selling domain. We have used Java programming language and JAFMAS framework architecture. JAFMAS is a Java – Based Framework for Multi Agent Systems, which provides coherent development of this kind of systems. JAFMAS was created by the researchers from Department of Electrical and Computer Engineering and Computer Science at the University of Cincinnati [6]. In our system agent communicate with other agents through direct announcements and through messages to all agents in multicast mode. When two agents (buyer and seller) start bilateral conversation, they switch from the multicast mode to the direct one. During the conversation, the agent can change the state of conversation thread depending on message content, meeting some conditions like state of other concurrent conversations or change of agent's goal. While constructing the answer the agent may choose from several available rules. The chosen rule has to be appropriate to the present state of conversation and to negotiation strategy. In our system negotiation mechanism is integrated with communication language. While negotiating, the agent can be in two active states: "trade" and "wait". "Accepted" and "rejected" are final states of the conversation (see Figure 4).

Hypothetical run of the negotiation thread may be as follows. Buying agent sends a message to the community of agents announcing that it wants to buy a specific car and waits for proposals. Then the seller may send a proposal message and the negotiation threads starts between these two agents. Each agent can run concurrently several threads. The negotiation thread is running to the moment, when one agent sends the approval to the other agent and waits for the commitment. To commit the deal we use two-phase commitment protocol.

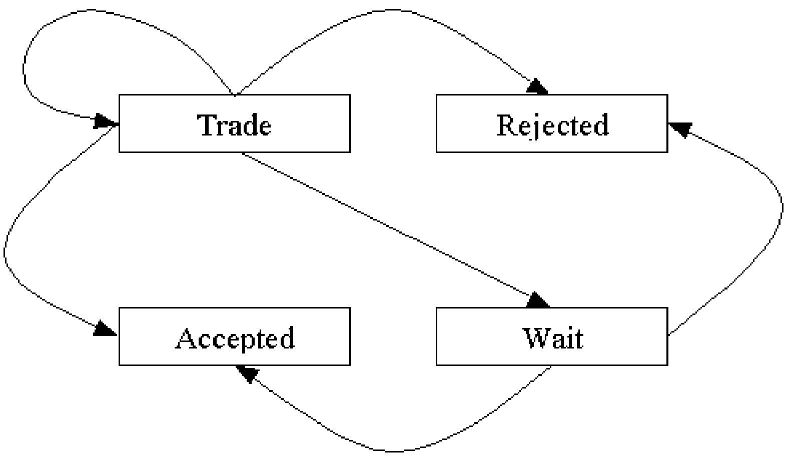


Fig. 4. Possible states of negotiation thread. The "Wait" state is necessary for two-phase commitment. It may also be transitional state between "Trade" and "Accept" or "Reject". When the thread is in "Wait" state, the other threads cannot end with success.

Each negotiation thread may concern many aspects. The buyer decides which aspects are important for him and such multi-aspect negotiation is run. There are two types of the aspects:

- Continuous – a set or range of possible values is infinite (e.g. price, delivery time),
- Discreet – a set of domain values is finite and rather sparse (e.g. colour, equipment).

Table 1. Examined aspects

Aspect name	Aspect type	Set (or range) of values
Price	Continuous	Range: [0$,1 000 000$]
Colour	Discreet	Set: {White, Green, Blue, Red, Black};
Model	Discreet	Set: {Fiat Stilo, Ford Focus, Opel Astra, Peugeot 307, VW Golf};
Equipment	Discreet	Set: {Standard, Comfort, Elegance}

We created three profiles using aspects presented in Table 1:

- P1 containing price and model,
- P2 containing price, model and equipment,
- P3 containing all of above listed aspects.

The experiment was conducted in each aspect profile for three types of various strategies:

- S1 is a pure linear strategy,
- S2 is a time-dependent strategy,
- S3 is a resource-dependent strategy.

In the Figure 5 we can observe the hypothetical run of the negotiation thread from agents' point of view.

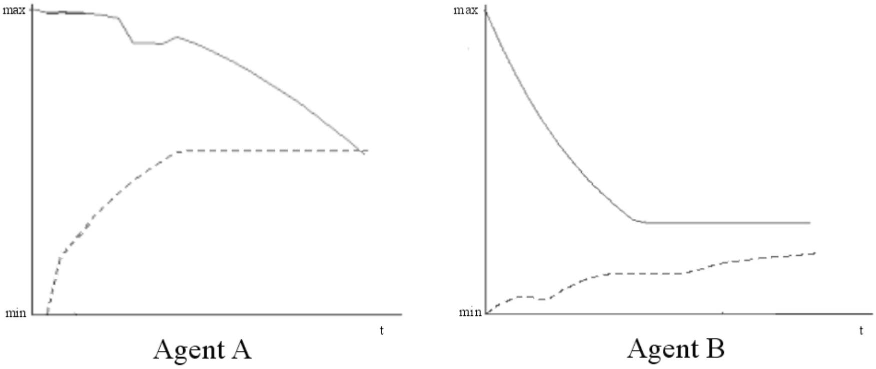


Fig. 5. Scoring function values in time from agents' point of view. Solid line denotes the scoring functions' values of own offers and dotted line denotes the values of other agent's offers Agent A reaches acceptable state earlier than the agent B, because the value of agent B offer is grater than the value of its own previous offer.

5.2 Experiment's Results

The complete results' description can be found in research report [11]. We have made the following remarks:

Remark 1. When increasing a number of aspects, the percent of successful negotiation threads increases. While the agents negotiate using more aspects they can better fit the other hand's offers.

Remark 2. When increasing a number of aspects, the standard deviation value of successful negotiation steps' number decreases. It is a result of more stable negotiation. The greater number of negotiation aspects causes the better awareness of negotiated subject.

Remark 3. When increasing a number of aspects, the number of negotiation steps logarithmically increases. The greater number of aspects results in increasing possibilities of changing agent's offer. It causes that negotiation lasts longer. Despite greater verbosity there is good idea to achieve better successfulness.

6 Conclusions

In this paper we have presented approach to analyse and develop bilateral, multi-aspect negotiations used in multiagent system. The role of negotiation is not only to resolve conflicts between agents, but also to enforce agents' social behaviour. Petri nets are very suitable formalism to describe concurrent systems without deadlocks. The experiment, which has been conducted, has shown that there is a strong relationship between variety and number of used aspects and effectiveness of negotiation. Despite decreasing efficiency we suggest using several aspects to achieve successful negotiation. The next step in research is to find *pareto* optimal solutions for negotiation threads.

References

1. Barber, K.S., Liu, T.H., Goel, A., Martin, C.E.: Conflict Representation and Classification in a Domain-Independent Conflict Management Framework. Proc. Third International Conference on Autonomous Agents, Seattle (1999) 346-347
2. Faratin P., Jennings N., Sierra C.: Negotiation Decision Functions for Autonomous Agents. Int. Journal of Robotics and Autonomous Systems 24 (1997) 159-182
3. Ferber, J.: Multi-Agent Systems. Addison-Wesley (1999) 12
4. Fernandes, J.M., Belo, O.: Modeling Multi-Agent Systems Activities Through Colored Petri Nets: An Industrial Production System Case Study. 16th IASTED International Conference on Applied Infomatics (AI'98), Garmisch-Partenkirchen (1998) 17-20
5. Ferraro, A., Rogers, E.H.: Petri nets in the evaluation of collaborative systems. IEEE Int. l Conf. on Systems, Man & Cybernetics (1997)
6. Galan, A.K.: JiVE: JAFMAS Integrated Visual Environment. A thesis submitted to the Division of Research and Advanced Studies of the University of Cincinnati, (2002)
7. Hiraishi, K.: PN^2 - An Elementary Model for Design and Analysis of Multi-agent Systems. (2002)
8. Ioerger, T.R., Xu, D., Volz, R.A., Yen, J.: Modeling and Analyzing Multi-Agent Behaviors Using Predicate/Transition Nets. International Journal of Software Engi-neering and Knowledge Engineering 13 (2003) 103-124
9. Jennings, N., Lomuscio, A., Wooldridge, M.: Agent-Mediated Electronic Commerce. Springer Verlag. (2000)
10. Lenar, M., Zgrzywa, A.: Modelling Negotiating Multi Agent Systems Using High Level Petri Nets. Information Systems Architecture and Technology Seminar ISAT 2001, Proceedings of the 23rd International Scientific School Digital Economy Concepts, Tools and Applications, Wroclaw (2001)
11. Lenar, M.,: Rozwlekło i skuteczno mechanizmu negocjacyjnego w bilateralnej negocjacji wieloaspektowej (in Polish). Research report of SPR Series No 33, Wroclaw University of Technology, Wroclaw (2002)
12. Purvis, M., Nowostawski, M., Cranefield, S., Purvis, M.: Multi-Agent System Interaction Protocols in a Dynamically Changing Environment (2002)
13. Zlotkin, G., Rosenschein, J.: A Domain Theory for Task Oriented Negotiation. International Joint Conference on Artificial Intelligence (1993) 416-422
14. Zlotkin, G., Rosenschein, J.: Compromise in Negotiation: Exploiting Worth Functions Over States. Artificial Intelligence, Vol. 84 (1996)

Multi-agent Development Toolkits: An Evaluation

Elhadi Shakshuki and Yang Jun

Jodrey School of Computer Science
Acadia University, Wolfville, NS, Canada
{elhadi.shakshuki;052982y}@acadiau.ca

Abstract. Today, multi-agent systems (MAS) play an important role in the information technology. To develop these systems requires the developer to deal with several issues and to implement many of the system components, such as protocols, name services and agents' functionalities. Although, there are several agent building toolkits that can help developers to effectively develop MAS and allow them to focus more on the application specific domains, it is difficult for developers to select the appropriate one. This paper provides an evaluation of Java Agent Development (Jade) framework, Zeus Agent Building Toolkit (Zeus) and JACK Intelligent System (Jack); with special focus on the following main criteria: Java support, performance evaluation, development support and performance on message transport system.

1 Introduction

Reinforcement learning Many multi-agent systems are developed for different domains in business and research sectors. Using multi-agent systems toolkits facilitates the process of developing such systems quickly and efficiently. But, there exist a large number of multi-agent toolkits [10,11], which makes it difficult to select an appropriate one. Therefore, it becomes necessary to analyze and compare these toolkits and help developers to choose the most appropriate toolkit. One of the main problems of these tools, they show significant differences among them in terms of design concepts, classes provided and performance. There have been few attempts by some researchers to compare these tools. For example, in [4] the focus was on evaluating a single multi-agent toolkit. The Agenticities project [7] is an initiative to deploy the worldwide test-bed of agent platforms and services based on FIPA agent standard. In this project, the main focus was however on function testing and not on performance testing. The EvalAgents project aimed at evaluating the agent building toolkits on Java support, mobility, security and development support area [5]. In another direction, Camacho et al. [2] performed experiments to test and compare the performance on Zeus, Jade and Skeleton-agent frameworks, using the meta-search engine workload. Although, they performed useful experiments there was little work has been done on the performance measure of the message transport system of toolkits. We strongly believe that one of the main key components of the toolkits is the message transport system. We also believe that the second most important and challenging issue in multi-agent systems' platforms is the transport

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 209–218, 2004.

layer for agent communication [4]. Towards this end, we developed a benchmark to compare Jade, Zeus and Jack agent building toolkits performance on the message transport system (MTS). We implemented a prototype of a sample system using each of these toolkits to measure their performance. The evaluations of these toolkits were performed on a notebook pc with Intel Pentium 4 1.6 GHz processor, 256 MB RAM with a JDK running Windows XP.

2 A Benchmark Description

Several A set of evaluation categories to evaluate multi-agent toolkits was described in [12]. Communication that provides the languages and protocols to exchange messages between the agents and the scalability that provides the system's ability in handling more agents are the key features of multi-agent systems. One appropriate performance measure is to calculate the time required to send and receive a message between agents. We developed a benchmark with the capability to calculate the average roundtrip time (*ART*) that is required for a message to travel from an agent (called it sender) to another agent (called it receiver) and back to the sender. This can be described formally as in equation (1):

$$\mathrm{ART} = (t_j - t_i)/\mathrm{m}\ . \tag{1}$$

Where, t_i refers to the initial time when message *i* is sent by the sender (i.e., start time), t_j refers to the time when message *j* is received by the sender (i.e., end time), *m r*efers to the number of messages sent during the time that laps between t_j and t_i, and the unit of ART is millisecond.

The messages' workflow between the sender and receiver agents can be described as follows. First, the sender formulates a message and sends it to the receiver. At the same time an alarm is activated to record the starting time. Meanwhile, the sender waits for the arrival of the message. As soon as the receiver agent receives the message it sends it back to the sender. In our experiments, the sender agent sends up to 4000 messages. When the sender reaches this number; it records the time as the end time of sending messages. In order for an agent to calculate the number of messages sent and received per second, *M*, the following equation (2) is used.

$$\mathrm{M} = 1000/\mathrm{ART}\ . \tag{2}$$

Java method long System.currentTimeMillis() is used to measure the time requires the benchmark to divide the result by 1000 to convert the time into milliseconds. The benchmark tests the scalability of each toolkit by monitoring the performance of the toolkit, while the number of couple (sender and receiver) agents is increased. During the tests, the i^{th} sender communicates with i^{th} receiver by exchanging 4000 messages. To make the comparisons and calculations more specific, each message' content field is filled exactly with a string of 7 characters, similar to that in [3].

The agent creation/destruction degrades the system performance and affects the value of the performance matrix that we are interested in. Therefore, three intervals are identified while the benchmark is running, as shown in Figure 1. In this figure, Na(t) refers to the number of active agents and *N* is the total number of agents. The solid line shows the number of agents that are actually sending messages. The dotted line shows the number of created agent. During T_1 all agents are sequentially created and started utilizing the CPU time, and after a short period of time (solid line) they began to exchange messages. In this phase, we observed that the measurements are distorted due to the following reasons: the measured *ART* tends to get lower (i.e., faster exchange) because not all couples are created yet; the *ART* tends to appear higher (i.e., slower exchange) because the agent's creation takes CPU time. During the interval T_2, all couples are created and are ready to exchange messages. Within the interval T_3, the measurement is again influenced by the lower number of agents competing for the CPU time and by the agent destruction time. Similar behaviour is observed during the interval T_1. In our experiments, we focused on the messaging subsystem performance without considering the effects on *ART* [3].

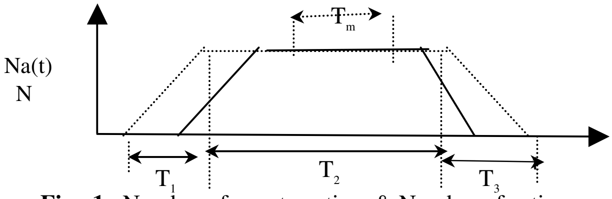


Fig. 1. Number of agent vs. time & Number of active agent vs. time

The time interval T_m identifies that the system is under 'full load'. Thus, the benchmark measures the system performance during this time. During this interval, the number of messages exchanged between the agents starts at 2000 messages and ends with 6000 messages. According our repeatedly test, we know that when one agent has exchanged 2000 messages, other agents are all working on exchanging message which means the system is under 'full load'. In order to get more precision result to reflect MAS real performance on MTS, 4000 messages have to be exchanged during measuring time. The timing system calls in Java are also considered as a major effect on the system total performance. In order to minimize this effect on the *ART*, only one agent is assigned the function of recording the timing when the benchmark is running. On the other hand, other agents can only receive/send messages and run as a workload simulator that produces system stress. The proposed performance matrix consists of three fields, namely: the number of active agents, the average message round trip time, and the number of sent and received messages per second.

3 Implementations and Experimental Results

3.1 JADE

Jade (Java Agent Development framework) is a software framework that facilitates the development of multi-agent systems in compliance with the FIPA specifications.

Jade is a developed by the research institute of Telecom Italia, CSELT S.P.A. The platform is available in version 3.0b1 as of March 2003. It is an open source and can be downloaded for free from Jade website [6]. Jade supports Java for development and execution. The Jade platform supplies some tools for graphical administration and application development. Jade is a distributed agent platform, which can be split among several hosts. One Java application is implemented and therefore only one java virtual machine is executed on each host. Agents are implemented as Java threads and live within agent containers that provide the runtime support to the agent execution. Jade supports intra-platform agent mobility, including transfer of both the state and the code of the agent. It also supports the execution of multiple, parallel and concurrent agent activities via the behaviour model.

The following provides a detailed description of Jade performance on the message transport system (MTS). In our experiments, two cases are considered. In the first case, the communication between agents is established when both agents are living in the same container. In the second case, the communication between agents is established when agents are living in different containers. Each JVM is the basic container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute their tasks on the same host [3].

Table 1 shows the performance results when all agents run in the same container. These results reveal that Jade has good performance on the message transport system when agents are living in the same container.

Table 1. MTS Performance results for Jade on one container

Number of active agents	ART (millisecond)	Number of messages (per second)
2	0.453	2222
4	0.803	1250
8	1.648	606
16	3.5775	279
32	8.955	112

Figure 2 shows the change of *ART* and the number of messages sent and received with respect to the increase in the number of agents. These figures demonstrate a linear relationship between the time required for the message to travel back and forth, and the number of increased couple agents. When the number of active agents increases, it requires more threads to run. Consequently, more agents became competing for CPU time; thus the performance decreases.

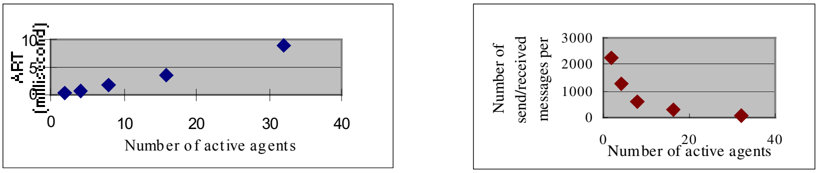


Fig. 2. MTS Performance results for Jade on one container

Table 2 shows Jade performance on the message transport system: agents are living in two containers is poorer than that when agents are living in the same container.

Table 2. MTS Performance result for Jade on two containers

Number of active agents	ART (*millisecond*)	Number of Messages (*per second*)
2	12.75	78
4	22.74	44
8	38.70	26
16	107.37	9
32	243.9	4

The change of *ART* and the number of messages sent/received with respect to the increase in number of agents, when the agents live in two different containers, is shown in Figure 3. Although, there is no clear relationship that can be derived in this case, we can observe that as the number of active agents increases require more threads to run and compete for CPU time that made the performance to get lower.

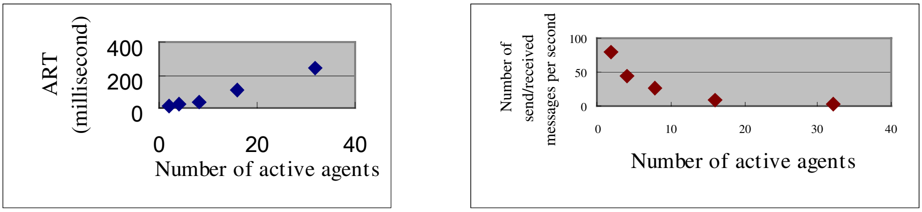


Fig. 3. MTS Performance result for Jade on two containers

Jade uses different mechanism to transport messages between agents that live in the same containers and agents that live in different containers. This results into a performance differences. Jade optimizes on agents localization and uses event passing when the agents are in the same containers. However, when the communication happens between agents on two different containers, Jade uses RMI to send the messages and has lower performance compared to that on one container.

3.2 ZEUS

Zeus agent building toolkit is a toolkit for constructing collaborative multi-agent applications. Zeus is developed at the British Telecom (BT) Labs' intelligent systems research group. The platform is available in version 1.03b as of May 2000. It is an open source and can be downloaded for free from the website [9]. Zeus supports Java for development and execution. Zeus has an agent building environment, a suite of tools that facilitate the construction of multi-agent systems. It generates the executables for the agents automatically. Zeus is a FIPA-compliant platform. All Zeus agents communicate using messages that obey the FIPA 1997 ACL specification. Zeus provides the agent with a component library, which allows the agents to communicate and coordinate with other agents.

The same experiments performed on Jade are also performed on the Zeus toolkit, which include the two cases mentioned previously. The performance results when all agents run in the same container are as shown in Table 3. These results show that there is no much difference between 2 couple and 4 couples of agents.

Table 3. MTS Performance result for Zeus on one container

Number of active agents	ART (millisecond)	Number of messages (per second)
2	100.945	10
4	103.5	10
8	149.9	7
16	264.7	4
32	519.375	2

The relationship between the number of agents and *ART* and the number of messages sent and received are shown in Figure 4. Consider that these results are recorded when the agents are in one container. We observe from these figures that when 4 couples of agents are running, the total processing time reaches up to 45%. While, when 8 couples of agents are running, the total processing time reaches up to 90%. This is again makes it clear that when the number of active agents increases, more threads are required to run and compete for CPU time. Thus, the performance becomes lower.

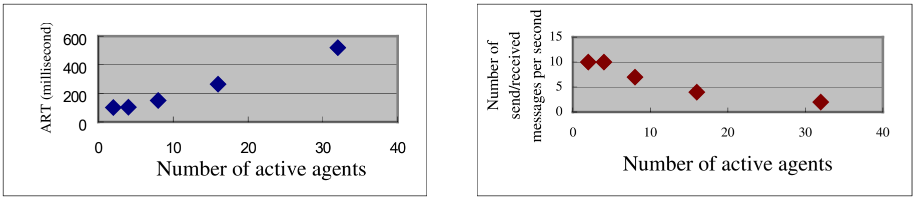


Fig. 4. MTS Performance result for Zeus on one container

The performance results for Zeus on the message transport system when agents are living in two different containers are shown in Table 4. These results reveal that there is only a slight difference in performance between agents when they live in the same container and agents when they live in different container.

Table 4. MTS Performance result for Zeus on two containers

Number of active agents	ART *(millisecond)*	Number of messages (per second
2	100.84	10
4	101.77	10
8	123.7	8
16	332.9	3
32	680.2	1

The relationship between the number of agents and *ART* and the number of messages sent and received are shown in Figure 5. The communication mechanism is the same as for agents on one container or on different containers. Each agent has a mailbox component that implements its communication mechanism. The mailbox is used for creating and reading TCP/IP sockets for sending and receiving messages. The performance measured in the case of two containers is lower than that of one container. This is because of the fact that two JVMs consume more processor time than one JVM.

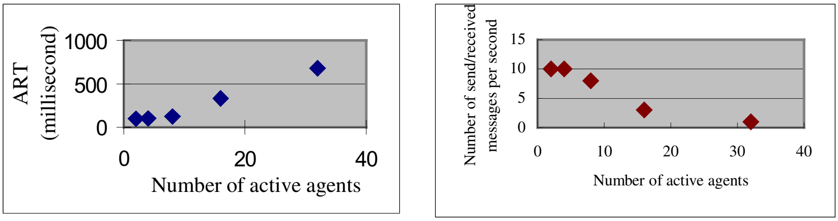


Fig. 5. MTS Performance result for Zeus on two containers

3.3 JACK Intelligent Agents

Jack intelligent agent is a commercial agent oriented development environment. It is a product of Agent Oriented Software Group and the platform is available in version 4.1. It should be noted that it is not an open source, however there is a trial version for 60 days that can be downloaded from this website [8]. Jack is a development environment that is built on top of Java and acts as an extension of Java that offers classes for implementing agent behaviour. It provides GUI for defining agents within projects. The GUI allows the developers to modify the agents' views, belief sets, capabilities and plans. Jack also contains an object browser (JACOB) that provides object modeling for communication of objects between agents and inputting of agents [1]. Jack intelligent agents are autonomous software components that have explicit goals to achieve or events to handle according the theoretical Belief Desire Intention (BDI) model of artificial intelligence. In addition, it supports local communication and remote communication with different mechanism. Jack does not conform to FIPA standard.

We performed the same experiments, which includes the two cases. The performance results when all agents run in the same container are shown in Table 5.

Table 5. MTS Performance result for Jack on one container

Number of active agents	ART (millisecond)	Number of messages (per second)
2	1.675	597
4	3.08	325
8	6.07	165
16	11.9	84
32	24.08	42

Figure 6 shows the change of ART and the number of messages sent and received with respect to the increase in the number of agents. We can observe from these results that there is a linear relationship between the time required for the message to take a roundtrip and the number of couple of agents increased.

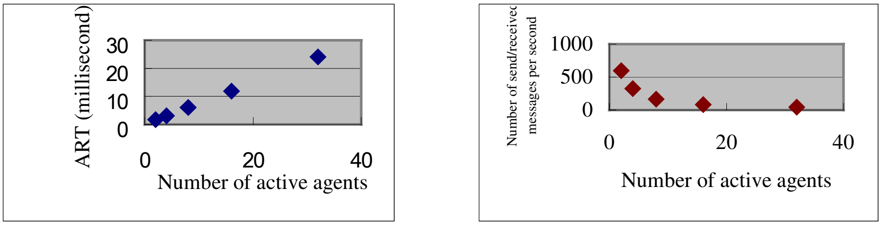


Fig. 6. MTS Performance result for Jack on one container

Table 6 shows the performance of Jake on the message transport system when agents are living in two different containers is poorer than that when agents are living in the same containers.

Table 6. MTS Performance result for Jack on two containers

Number of active agents	ART (*millisecond*)	Number of Messages (*per second*)
2	4.34	230
4	8.24	121
8	14.77	68
16	29.89	33
32	59.51	17

Figure 7 shows the change of *ART* and the number of messages sent and received with respect to the increase in number of agents; when the agents live in two different containers. We observe from these results that there is a linear relationship between the time required for the message to take a roundtrip and the number of couple of agents increased. ART approximately doubles as the number of active agents doubles.

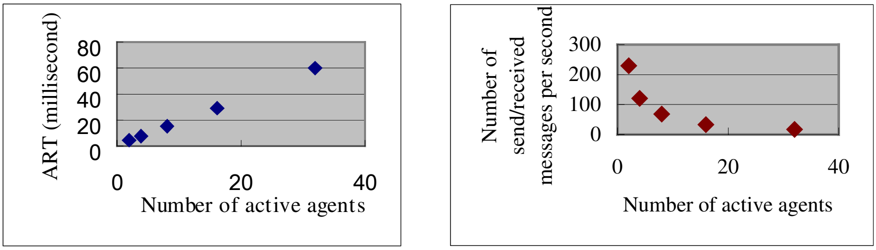
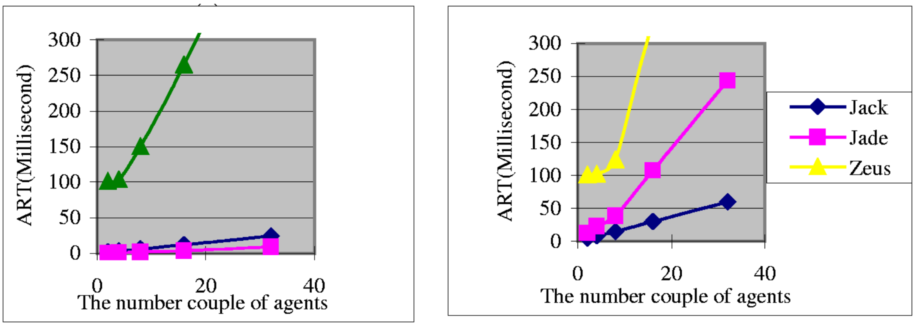


Fig. 7. MTS Performance result for Jack on two containers

Jack uses different mechanisms to transport messages between agents that live in the same containers and agents that live in different containers. This results into performance differences. When agents share the same container, the routing of messages between them is trivial. In this case, the sender agent only needs to know the receiver agent's address to send the message accordingly. When agents are running in the different containers, Jack uses DCI network as communication layer to allow agents to communicate. This network layer allows agents to communicate through ports using a Jack transport protocol; UDP with guaranteed reliability. This makes it clear that Jack has lower performance on two containers as compared to that on one container.

4 Comparison

This section provides a comparison between the three agent building toolkits including Jack, Zeus and Jade. First, all of these toolkits provide Java support. Only Jade and Zeus are available for free with reasonable documentation that developers can use. Zeus and Jade are FIPA-compliant agent platforms. Jack does not make use of any pre-existing standard agent communication language. Jade uses an agent model and Java implementation that offer good runtime efficiency and software reuse. Conversely, Zeus offers very powerful means to graphically design multi-agent systems, with the desired interactions. Jack provides an excellent GUI for defining agents within projects. Jack includes all components of the Java development environment as well as offering specific extensions to implement agent behaviour. Figure 8-a shows the performance results of the three toolkits when all agents live in the same container. From these results we draw the following conclusions: Jade provides better performance on the message transport system than Jack and Zeus.



(a) (b)

Fig. 8. (a) Performance results for Jade, Jack and Zeus on one container of MTS, (b) Performance results for Jade, Jack and Zeus on two containers of MTS

Figure 9 shows the performance results of the three toolkits when the agents live in different containers. From these results we draw the following conclusions: Jack provides better performance on the message transport system than Jade and Zeus.

5 Conclusions and Future Work

Agent building toolkits facilitate the process of developing multi-agent systems. This paper compared Jade, Zeus and Jack multi-agent toolkits and proposed a benchmark to evaluate them. The implementations of sample tests of the multi-agent systems for the benchmark have demonstrated how different toolkits might support the developers. The agent building toolkits provide their own architecture and build-up methodology to deploy multi-agent systems. Based on our investigations, it is recommended that Jade can be used when the application requires agents to live in the same container, because it provides better performance. Alternatively, it is recommended using Jack when the application requires agents to live in different containers.

In the future, we will continue to work on other agent building toolkits, using the same test measurements as well as incorporating other important issues. Furthermore, we will design and implement user friendly interfaces that allow developers to find the most appropriate toolkits based on their required needs.

References

[1] Bitting, E., Carter,J. and Ghorbani,A.A., “Multiagent System Development Kits: An Evaluation”, In the Proceedings of the 2003 Communication Networks and Services Research Conference, May 15-16, Moncton, New Brunswick, Canada, 2003.

[2] Camacho, D., Aler,R., Castro,C. and Molina.M.J, “Performance Evaluation of Zeus, Jade, and Skeletonagent Frameworks”, 2002.

[3] Cortese, E., ‘‘Benchmark on Jade Message Transport System”, 2002.

[4] Fonseca,S., Griss,M. and Letsinger,R., ‘‘Evaluation of the Zeus MAS Framework’’, Software Technology Laboratory HP Laboratories Palo Alto, HP Labs Technical Report, HPL-2001-154, 2001.

[5] Grabner, M.,Gruber,F., Klug,L.,Stockner,W., Altmann,J. and Essmayr,W., “Technical Reprot: SCCH-TR-00-64-1: EvalAgents: Evaluation Report”, 2000.

[6] http://Jade.cselt.it.

[7] http://www.agentcities.org.

[8] http://www.agent-software.com

[9] http://www.labs.bt.com/projects/agents/Zeus.

[10] MultiAgent Systems, Tools for building MASs. http://www.multiagent.com/Software/Tools_for_building_MASs/index.html.

[11] Reticular Systems Inc, List of Agent Construction Tools, http: //www.agentbuilder.com/ AgentTools.

[12] Shakshuki, E., Koch, L. and Kamel, M., ‘‘Evaluation of Multi-Agent Toolkits: A Benchmark’’, The 2003 International Conference on Artificial Intelligence (IC-AI'03), pp. 797-802, Las Vegas, USA, 2003.

An Artificial Immune System for Fault Detection

Jose Aguilar

CEMISID, Dpto. de Computación, Facultad de Ingeniería, Av. Tulio Febres. Universidad de los Andes, Mérida 5101, Venezuela
aguilar@ing.ula.ve

Abstract. The oil well instrumentation generates a set of process variables, which must analyzed by the experts in order to determine the well state. That implicates a highly cognition task where the information generated is very important for maintenance tasks, production control, etc. In other way, the natural energy of an oil field can not be enough to lift the fluids. In these case is necessary to use another procedure to lift the oil, for example gas. That is an interesting case to be modeled by an artificial intelligence technique. Particularly, in this paper we propose an Artificial Immune System for fault detection in gas lift oil well. Our novel approach inspired by the Immune System allows the application of a pattern recognition model to perform fault detection. A significant feature of our approach is its ability to dynamically learning the fluid patterns of the 'self' and predicting new patterns of the 'non-self'

1 Introduction

When the natural energy of an oil field is not enough to fluid lift, we need a secondary recovery procedure on the well (normally, this is called artificial lift) [2, 12]. For this case, we can use different techniques such as artificial lift by gas (ALG) [2, 12]. The idea of the ALG is to inject gas, and in this way to lighten the fluid. The design of this system must be made very carefully. One of the aspects to consider is the fault detection [2, 4]. On the other hand, the immune system is a collection of cells and organs able to perform tasks with characteristics such as pattern recognition, learning, noise tolerance, distributed detection, and memory, with the purpose of maintaining the physical integrity of an individual [8, 11, 13, 15, 20]. The problem that the immune system solves may be described as the distinction between self and non-self entities, being the self entities the internal cells and molecules produced by the body, while the non-self entities correspond to potentially harmful foreign entities such as viruses, parasites and bacteria. In recent years, some immunity based computational models have been successfully developed [3, 4, 6, 8, 9, 10, 14, 15, 17, 18, 21], showing an enormous potential for practical applications to other fields such as computer security and pattern recognition [6, 7, 14, 16, 19]. In this paper we discuss an immunocomputational framework to define a fault detection system for a gas lift oil well. Particularly, we propose a fault detection system based on immune system ideas for gas lift oil well.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 219-228, 2004.

2 Theoretical Aspects

2.1 Artificial Immune Systems

In the recent years, a novel approach has begun to emerge which is the use of concepts from immunology to solve problems [8, 11, 13, 15, 20]. The human immune system has a very distributed and adaptive, novel pattern recognition mechanism. The body recognizes its own cells from those of the invaders [20]. In general, the purpose of the immune system is to protect the body against infection and includes a set of mechanisms collectively termed humoral immunity. The immune system uses learning, memory, matching, diversity, distributed control and associative retrieval to solve recognition and classification tasks [20]. In particular, it learns to recognize relevant patterns, remember patterns that have been seen previously, and uses combinatorics to construct pattern detectors efficiently. The immune system also remembers successful responses to invasions and can re-use these responses if similar pathogens invade in the future. Matching refers to the binding between antibodies and antigens. Diversity refers to the fact that, in order to achieve optimal antigen space coverage, antibody diversity must be encouraged. Cloning and hypermutation maintain the diversity of the antibody set. Distributed control means that there is no central controller, rather, the immune system is governed by local interactions between cells and antibodies. The antibodies are present through out the body without any central control and thus defend the body by this interaction in a distributed fashion. These remarkable information-processing abilities of the immune system provide several important inspirations to the field of computation [8, 11, 13, 15, 20].

There are many more features of the immune system, including adaptation, idiotypic network and protection against auto-immune attack. The immune system must maintain a diverse repertoire of responses because different pathogens must be eliminated in different ways. To achieve this, the immune system constantly creates new types of responses. These are subject to selection processes that favour more successful responses and ensure that the immune system does not respond to self-proteins. Lymphocytes are subject to two types of selection process. Negative selection, which operates on lymphocytes maturing in the thymus (called T-cells), ensures that these lymphocytes do not respond to self-proteins. The second selection process, called clonal selection, operates on lymphocytes that have matured in the bone marrow (called B-cells). Any B-cell that binds to a pathogen is stimulated to copy itself. The copying process is subject to a high probability of errors ("hypermutation"). The combination of copying with mutation and selection amounts to an evolutionary algorithm that gives rise to B-cells that are increasingly specific to the invading pathogen.

2.2 Gas Lift Well

When we search oil (exploration), we need to use scientifique methods to determine the subsoil characteristic. In this way, we can know if there is a region with an accumulation of hydrocarbon. When the hydrocarbon has been detected, the exploitation of the oil field starts. It consists in bringing the oil to the surface using the natural energy of the oil field or other methods (for example gas artificial lift). Normally, at the beginning the natural energy of the oil field allow the oil lift, but when the oil field is old we need to use other techniques: gas artificial lift, mechanical pump, hydraulics pump, and so forth. The gas artificial lift technique is a technology based on the injection of gas to allow the fluid of the oil to go to the surface. The gas comes from compression plants, through a gas distribution system. The last part is composed by gas multiples (MLAG) and pressure high multiple (MAP). In general the gas goes to the oil well, and its injection is controlled by control equipment that is in the surface and subsoil. We need to inject the optimal quantity of gas to obtain the minimal pressure to allow the fluid lift. During the perforation, there is a cementation phase to glue a tube called 'casing'. Inside of it we include another tube called 'tubing'. This last tube is used to transport the oil from the oil field to the surface [2]. With the data from the pressure table of the "Casing" and "Tubing" we can determine the next information [2, 4]: 1. Surface Restriction: a high pressure of the "Tubing"; 2. Freezing: a fault in the gas injection or small quantity of recovery fluid due to the freezing of the tubes; 3. Sandy or coat well: not continuos lift; 4. Frequent cycles of very fast lift: small pressure of the "Casing"; 5. Far cycles with intermittent lift: small fall of the pressure of the "Casing"; 6. Valve work bad: fall and climb of the pressure of the "Casing"; 7. Valve does not work: the pressure of the "Casing" is smaller than the pressure of operation of the valve.

3 Our Fault Detection System

This section introduces our fault detection system based on the AIS. In essence, the immune system is used here as inspiration to create an unsupervised machine-learning algorithm. We develop a detection mechanism by maintaining immune cells that detect an anomaly. Our approach works in two phases: at the beginning is generated the lymphocyte (outline operation phase). Then, our system is included in the environment (inline operation phase) to detect the anomalies.

3.1 Outline Operation Phase

This phase is based on the negative selection algorithm to generate the B and T lymphocytes, where the cells that reactions with the self organism are eliminated. The macro-algorithm is: i) Recollection of data in normal state. ii) Pre-processing of data. iii) Representation of data. iv) Save the "self cells". v) Generation of detectors.

Recollection of Data in Normal State

We use the normal and abnormal condition patterns presented in [2, 4] like reference model. Each pattern is a set of registers "Tubing-Casing" (see Figure 1). Some of the diagnostic with these patterns are: i) ***Norma Operation***: small variation of the "Tubing" and "Casing" pressure (Figure 1.a); ii) ***Low Production***: The "Tubing" pressure increases and the "Casing" pressure is stabled (Figure 1.b); iii) ***Emulsion***: the "Tubing" and "Casing" pressures have small opposite tip; iv) ***Freezing Gas***: high variations of the “Tubing” pressure and the "Casing" pressure is stabled.

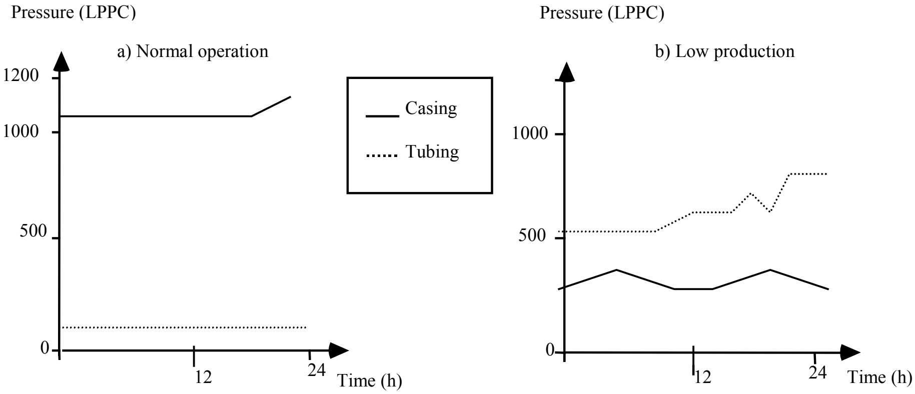


Fig. 1. Registers of the Tubing-Casing Pressures

For this phase, we have used the normal operation pattern (see Figure 1.a) to build the patterns to be used by the IAS.

Pre-processing of Data

We have defined a new representation of data, because our system uses information from different wells. In this way, we can unify the scale of the input values into the interval [-1,1]. We define a matrix (D[PxC]), where P is the number of data of each pattern (each data is a couple “Casing – Tubing” pressure) and C are the variables (in our case C=2, Casing and Tubing pressures). We use 80% of the normal condition patterns for generation of the detectors and 20% to test our system (to detect abnormal condition operations). The maximum and minimum value of each variable is determined using the set points of them. The parameters α and β define the fraction above or below of the set point that determine the maximum and minimum of each variable, respectively. Then, the transformation for each data of D is:

$$D^{k}{}_{ji} = 2\times\left(\frac{D^{k}{}_{ji} - \min^{k}{}_{i}}{\max^{k}{}_{i} - \min^{k}{}_{i}}\right) - 1 \qquad (1)$$

where:

$D^{k}{}_{ji}$ is the value of the row j, column i, of the pattern k modified. with: $j \in (1,2,\ldots P)$; $i \in$ (1,2); $k \in$ (1..3)

$\max^{k}{}_{i}$ is the maximum value of the variable or column *i* of the pattern *k*:

$$[\max^{k}{}_{i} = Op^{k}{}_{i} + \alpha_i \times Op^{k}{}_{i} \qquad 0 \le \alpha_i \le 1 \ ; \quad i \in (1,2) \ ; \ k \in (1..3)\ ].$$

$\min^{k}{}_{i}$ is the minimum value of the variable or column *i* of the pattern *k*:

$$[\min^{k}{}_{i} = Op^{k}{}_{i} - \beta_i \times Op^{k}{}_{i} \qquad 0 \le \beta_i \le 1 \ ; \quad i \in (1,2) \ ; \ k \in (1..3)\ ].$$

$Op^{k}{}_{i}$ is the set point of the variable *i* of the pattern *k*.

The value of α and β for each variable are:

$\alpha_{Casing} = 0.15$ $\beta_{Casing} = 0.15$

$\alpha_{Tubing} = 0.5$ $\beta_{Tubing} = 0.5$

Representation of Data

With D, we have built a new vector of representation of data (X) over each variable using a sliding window with size ($2 \le l < P$) and sliding length ($1 \le shift < P$).

$X^{k}{}_{1} = \lfloor D^{k}{}_{11}\ D^{k}{}_{21}\ \ldots\ D^{k}{}_{P1} \rfloor$ is a vector with all the elements of column 1 of D^k, and $X^{k}{}_{2} = \lfloor D^{k}{}_{12}\ D^{k}{}_{22}\ \ldots\ D^{k}{}_{P2} \rfloor$ is a vector with all the elements of column 2 of D^k.

We build sliding windows for each vector $X^{k}{}_{i}$, with $i \in (1,2)$. Thus, we generate a matrix for each variable *i* of each pattern *k* where the number of rows represents the numbers of sliding windows generated from vector $X^{k}{}_{i}$:

$$M^{k}i = \begin{bmatrix} X^{k}{}_{i1} \\ X^{k}{}_{i2} \\ \vdots \\ X^{k}{}_{ih} \end{bmatrix}$$

where: $M^{k}{}_{i}$ is the matrix of the variable *i* of the pattern *k*, and its elements are row vectors which represent the sliding windows generated from $X^{k}{}_{i}$.

$X^{k}{}_{ij}$ is the window *j* of the variable *i* of the pattern *k*. Where j ∈ (1,....,*h*) and *h* is the number of sliding windows generated. The size of each vector $X^{k}{}_{ij}$ is *l*.

The window is moved according to the sliding length to build a second window. When the sliding window matrix is built for each variable *i* of each pattern *k,* we unify the vectors that represent the same position of each variable for each pattern *k*. Now this matrix is called *V* with size h x (2 x *l*) for each pattern *k*. For example, the first vector of pattern *k* is:

$$V^{k}{}_{1} = \lfloor D^{k}{}_{11}\ \ D^{k}{}_{21}\ \ldots\ D^{k}{}_{l1}\ \ D^{k}{}_{12}\ \ D^{k}{}_{22}\ \ldots\ D^{k}{}_{l2} \rfloor$$

Then we determine the angle between contiguous data, and between the last element and the first element of this vector. That is, for the first vector of the pattern *k* we determine the angle as:

$$\tan(A^{k}{}_{1,1}) = {D^{k}{}_{21}}\Big/{D^{k}{}_{11}} \qquad (2)$$

and the last angle is determined as:

$$\tan(A^{k}{}_{1,2\times l}) = {D^{k}{}_{11}}\Big/{D^{k}{}_{(2\times l-1)1}} \qquad (3)$$

Where: $A^{k}{}_{g}$ is the angle matrix of the pattern *k,* for the vector *g*, $\mathrm{g} \in (1,\ldots,\mathrm{h})$ and $\mathrm{k} \in (1,\ldots,\mathrm{n})$

$$A^{k}{}_{g} = \left\lfloor A^{k}{}_{g,1} \cdots\cdots A^{k}{}_{g,2\times l} \right\rfloor$$

Save the "Self Cells"

The "self" is the set of sliding windows represents by the angles obtain previously. This, the self set is represented by the *k* matrices of the angles of size h x (2x *l*), where the elements are angles inside of the interval [0, 2xΠ radians].

Generation of Detectors

Our system uses the negative selection algorithm. That is, our AIS uses the self cells sets to produce detectors with the capabilities to discriminate the self and non-self. Each detector is an angle string generated randomly, which is defined as valid if it does not mate with the self (we avoid false positive). Now, we are going to present the negative selection algorithm used in this work, called the *Random generation of detectors*: We generate vectors (detectors) where theirs components are random angles inside of the interval [0, 2xΠ radians]. The number of detectors to generate and the coupling interval of the angle are given by the users. We search detectors that don't active them front the self set [3]. That is, we generate vectors (detectors) with random angles $B_1 = \lfloor B_{1,1} \quad B_{1,2} \cdots\cdots B_{1,2\times l} \rfloor$, and we compare them with all strings of the self set ($A_1 = \lfloor A_{1,1} \quad A_{1,2} \cdots\cdots A_{1,2\times l} \rfloor$). Each component of the detector B_1 has a sweep ΔB_1, and there is a coupling if:

$$\mathrm{B}_{1,\mathrm{i}} \leq \mathrm{A}_{1,\mathrm{i}} \leq \mathrm{B}_{1,\mathrm{i}} + \Delta\mathrm{B}_1 \qquad \text{with } 1 \leq \mathrm{i} \leq 2\times \mathrm{l}; \qquad (4)$$

If a detector couples a string of the self set, then we eliminated this detector, we increase the counter of deleted detector and we generate another detector. We finish with a given sweep when a given number of detectors deleted is achieved. In this case, we increase the angle sweep, we reset the counter of deleted detectors and we restart the random generation of new detectors. When we achieve a number given of pre-defined detectors, then we stop the procedure of generation of detectors.

3.2 Online Operation Phase

In this phase the immune system must identify the self cells. It is composed by the next tasks:

Recollection of New Data
We test our system using the patterns of the work [2, 4].

Pre-processing of Data
We use the same procedure of the previous phase.

Representation of Data
We represent the data using vectors build according to sliding windows of size *l* and the running length *shift* for each variable, then we use the same procedure like the previous phase to obtain vectors with component which are angles inside of the interval [0, 2xΠ radians].

Define if the Detectors Generated in the Previous Phase Can Identify or Not This New Data
The patterns represent abnormal conditions (antigens). Using each pattern of abnormal condition, we test the generated detectors in the previous phase. In this case, we need to test the coupling between them using a procedure similar than our detectors generation algorithm.

4 Experiments

Because the Tubing and Casing registers have small opposite peak at the same time, we use that to divide a fault pattern in 11 sections. For the case of intermittent injection, we divide this pattern in 7 sections that represent the fault regions by gas deficiency (Sections 1, 3, 5 y 7) and regions with quasi-stabilization (sections 2, 4 y 6). For freezing gas and low Production we divide the patterns in 4 sections. We follow a similar procedure to divide the rest of fault patterns (see [2, 4] for more details). The performance measures to evaluate our system are: a) *Execution Time (seconds):* is the CPU time of our system to generate the detectors. ii) Number of activated detectors by fault section. iii) Number of fault sections detected. We use the next parameters for our algorithms of detectors generation: a) Size of the sliding window (*l*) = {2, 3, 4}, b) Length of the sliding window (s*hift*) = 1, c) Number of detectors to generate (*num*) = {50000, 60000, 70000}, d) Coupling Interval (*asize*) = Π / 4.

The standard case is: *l*=2, shift=1, num=5000, and asize=Π/4. We have not modified *shift* because when we have modified these parameters we have not obtained important changes at the level of the result [4]. We have designed the next experiments for the outline operation phase: a) Different values of *l*, b) Different values of num. Now, we are going to show the main results, the rest of them can be seen in [4]. We have compared our algorithm with other work proposed in [4].

4.1 Variation of *num*

Table 1 shows the CPU time for each algorithm to generate the detectors for the low production fault pattern for different values of number of detectors. For the rest of case, the behavior is similar (see [4]). In general, if we increase the number of detectors, then the average of detection by fault section increase for both algorithms. The average of detection of our algorithm is bigger than algorithm [4]. We see in table 1 that the execution time of the algorithm [4] is smaller than our algorithm, but the number of fault sections detected by our algorithm is the biggest (see section 4.2).

Table 1: Execution Time for both Algorithms

		Execution Time (seconds)	
Parameters	Number of detectors	Our Algorithm	Algorithm in [4]
l = 2	50000	20605.77	4256.64
	70000	28226.438	36992.452
l = 3	50000	337249.01	120054.52
l = 4	50000	89538.891	52835.915

4.2 Variation of *l*

The table 2 shows the number of detectors activated for each algorithm in each fault section for the emulsion fault pattern for the Standard Case. The 11 sections of the pattern have been detected by our detectors. We can see more detectors activated by the algorithm 1. In [4] we present the rest of tables for the rest of fault patterns. In all case, our algorithm has a bigger number of detectors activated by section than the algorithm [4].

Table 2: Number of detectors generated by our algorithm and the algorithm proposed in [4] activated in each fault section of emulsion

Emulsion		Our algorithm	algorithm in [4]
Interval	Section size (minutes)	Number of detectors activated	Number of detectors activated
Section 1	50	12	2
Section 2	90	39	5
Section 3	130	39	3
Section 4	220	70	8
Section 5	90	55	2
Section 6	170	54	8
Section 7	120	39	3
Section 8	230	77	8
Section 9	100	50	2
Section 10	150	52	8
Section 11	90	31	3
Section average	130.909	47.091	4.727

5 Conclusions

This paper presented an artificial immune model specially designed to solve the fault detection problem for LAG well. Our system has demonstrated to be capable of combining exploitation with exploration and showed a good performance. This work has demonstrated that taking inspiration from the human immune system, in the form of the negative selection algorithm, is suitable for the design of novel error detection mechanisms. Error detection mechanism is probabilistic and performed in real-time, permitting a trade off between storage requirements and the ability to detect an error within the sequential system. In contrast to others error detection techniques that concentrate on single bit errors, and can sometimes fail to detect multiple errors, our immune system is adept to detecting this task. Particularly, we have generated detectors that determine deviation in the production process. This model can be used in system of high risk and real system, where we like to detect an abnormal condition operation very quickly. This model can be combined with other tools like a fault diagnostic system to classify the faults. In general, the model must be improved in: Add more dynamic (learning and memory systems) to avoid to use only the information catch outline; Use one algorithm to adapt the parameters of our system (coupling interval, size of the sliding window, etc.). Future work would benefit from other aspect of natural immune system: clonal algorithm, cell memories, etc.

References

1. J. Aguilar, M. Araujo, H. Aponte *"Fault Detection System in Gas Lift Well"*, *Proceeding of the International Joint Conference on Neural Networks*, pp. 1673-1677, 2003.
2. I. Albarran, "*Una Aplicación de Redes Neuronales en la caracterización del proceso de Levantamiento Artificial de Petróleo por Gas".* Technical Report, Universidad Central de Venezuela. Caracas, 2001.
3. M. Araujo, J. Aguilar, H. Aponte, "*Los Sistemas Inmunes Artificiales en problemas de Detección".* Technical Report, CEMISID, Universidad de los Andes, 2002.
4. A. Avizienis, "Towards Systematic Design of Fault-Tolerant Systems", IEEE Computer, Vol. 30:4, pp. 51-58, April 1997.
5. D. Dasgupta (ed), "Artificial Immune Systems and their Applications", Springer-Verlag, 1999
6. D. Dasgupta S. Forrest, "An Anomaly Detection Algorithm Inspired by the Immune System", in Artificial Immune Systems and Their Applications, Springer-Verlag, pp 262-277, 1999.
8. L. de Castro, J. Timmis, "An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm", Springer-Verlag, 2002.
9. L. de Castro, F. Von Zuben, "Immune and Neural Network Models: Theoretical and Empirical Comparisons", Int. Journal of Comp. Intelligence and Applications, Vol. 1:3, pp. 239-257, 2001.
10. S. Forrest, S. Hofmeyr, "Immunology as Information Processing," in Design Principles for the Immune Systems and Other Distributed Autonomous System, pp. 361–388. Oxford University Press, 2001.
11. S. Hofmeyr and S. Forrest, "Architecture for an Artificial Immune System," Evolutionary Computation, vol. 7(1), pp. 1289–1296, 1999.
12. A. Tarakanov and D. Dasgupta, "A Formal Model of an Artificial Immune System", BioSystems, Vol. 55: 1-3, pp. 151-158, 2000.

Heuristic Approach Based on Lambda-Interchange for VRTPR-Tree on Specific Vehicle Routing Problem with Time Windows

Naoto Mukai[1], Jun Feng[2], and Toyohide Watanabe[1]

[1]Department of Systems and Social Informatics,
Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603
{naoto, watanabe}@watanabe.nuie.nagoya-u.ac.jp
[2]Department of Information Engineering,
Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603
feng@watanabe.nuie.nagoya-u.ac.jp

Abstract. We propose VRTPR-Tree and a heuristic approach based on λ-interchange to solve a specific Vehicle Routing Problem with Time Windows (VRPTW). In the problem, delivery demands of customers are given as initial conditions. And, one of the vehicles with different positions visits the customers and transports to their destinations within time limits. For solving this problem, VRTPR-Tree indexes moving vehicles as a tree structure at some point. VRTPR-Tree generates an initial assignment condition for optimizing in a short time. An entry of a node consists of a pointer to a vehicle and a bounding rectangle which implies future positions of the vehicle (in leaf nodes) or pointers to child nodes and a bounding rectangle which encloses bounding rectangles of child nodes (in intermediate nodes). Initially, customers are assigned to a vehicle on the basis of the indexes of VRTPR-Tree, and the delivery orders of the customers are scheduled. Moreover, a heuristic approach based on λ-interchange optimizes the initial solution in the viewpoint of travel cost or customer satisfaction. We performed some experiments on an ideal environment. The experimental results show that our approach produces good results in short assignment and optimization times.

1 Introduction

In recent years, a new transportation system called demand-bus or dial-a-ride is focused by city governments. The traditional traffic system such as fixed bus system is one of the causes of traffic congestion in urban area. On the other hand, in the demand-bus system, customers could choose his riding and dropping points freely and are delivered by share-ride vehicle. This new traffic system enables to alleviate congested traffic conditions.

A Vehicle Routing Problem with Time Windows (VRPTW) is a basic model of demand-bus system. In VRPTW, plural demands of customers for delivery are given as initial conditions. And, each vehicle visits the customers and transports to their

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 229-238, 2004.

destinations within time limits. VRPTW has already been addressed in [1], [2], [3], [4]. Moreover, Solomon [5] introduced benchmark instances for VRPTW. Our problem is a specific case of the VRPTW. The major different is positions of vehicles, i.e. in most other papers, all vehicles leave from one depot, on the other hand, in our problem, the initial positions and velocity vectors of vehicles are different. Most of the methods for VRPTW used heuristic algorithms as well as our approach. However, their time complexities were too high because they tried to optimize in consideration of all vehicles and customers at once.

In our approach, VRTPR-Tree based on TPR-Tree[6][7] indexes moving vehicles spatially. VRTPR-Tree generates an initial assignment condition for optimizing in a short time. An entry of a node consists of a pointer to a vehicle and a bounding rectangle which implies future positions of the vehicle (in leaf nodes) or pointers to child nodes and a bounding rectangle which enclose bounding rectangles of child nodes (in intermediate nodes). The form of a bounding rectangle is based on a position and a velocity vector of a vehicle, road network constraints (i.e. reachable regions of a vehicle), and riding and dropping points of assigned customers. Initially, customers are assigned to a vehicle on the basis of the spatial indexes. And the delivery orders of customers are scheduled by FIFO (first-in first-out) queue. Moreover, our heuristic approach using λ -interchange optimizes the initial solution, and the optimal range of vehicles is restricted by the spatial indexes of VRTPR-Tree. The λ -interchange[8][9] is a basic algorithm for customer interchange between a set of vehicles.

The remainder of this paper is as follows: the formulation of VRPTW is described in Section 2. Section 3 defines the structure of VRTPR-Tree and bounding rectangles and time constraints. Initial solution of assigning vehicles and scheduling orders is defined in Section 4. An optimization algorithm using λ -interchange is defined in Section 5. Section 6 reports on our experimental results. Section 7 concludes and offers future works.

2 Formalization of VRPTW

We formulate VRPTW as follows. A traffic topology is based on the concept of a road network G which consists of nodes and edges. Nodes which represent intersections are given by Equation (1). A node is a pair of coordinates on x - y dimensions as (px, py). Edges which represent road segments between two nodes are given by Equation (2).

$$P = \{p_1, p_2, \cdots, p_l\} \tag{1}$$

$$L = \{[p - p'] : p, p' \subset P\} \tag{2}$$

Customers on the road network are given by Equation (3). The delivery demand of customer c_i consists of riding node, dropping node and time limit (i.e., the customer want to arrive at his dropping node within the time limit) as Equation (4). The customers could choose riding node and dropping node from the nodes in road network freely.

$$C = \{c_1, c_2, \cdots c_N\} \tag{3}$$

$$D_i = (r_i, d_i, TL_i) \tag{4}$$

The customer satisfaction of c_i is defined as Equation (5), where σ is control parameter of satisfaction. PC is the proportion of delay time to time limit TL, where td is dropping time instant. In particular, if a customer could arrive at his dropping node within his time limit, his customer satisfaction value is 1; otherwise, the value decreases gradually.

$$PC_i = \frac{td_i - TL_i}{TL_i}$$

$$CS_i = \begin{cases} 1 & (PC_i \le 0) \\ \exp\left(\frac{-PC_i}{\sigma}\right) & (PC_i > 0) \end{cases} \tag{5}$$

Vehicles are given by Equation (6). The position of vehicle v_j is given by Equation (7). The velocity vector of vehicle v_j is given by Equation (8).

$$V = \{v_1, v_2, \cdots v_K\} \tag{6}$$

$$\bar{v}_j(t) = (x_j(t), y_j(t)) \tag{7}$$

$$\vec{v}_j(t) = (vx_j(t), vy_j(t)) \tag{8}$$

The delivery order of vehicle v_j is given by a queue q_j in Figure 1. The riding nodes and dropping nodes are inserted into the queue in FIFO (first-in first-out) order. The traveling distance of the queue, which is the total distance of the route, is defined as Equation (9), where $d(p_1, p_2)$ is route distance between p_1 and p_2, and L is the length of the queue.

$$|q_j| = d(\bar{v}_j, q_j[0]) + \sum_{l=0}^{L-1} d(q_j[l], q_j[l+1]) \tag{9}$$

Consequently, our objective is to maximize the customer satisfaction and to minimize the traveling distance.

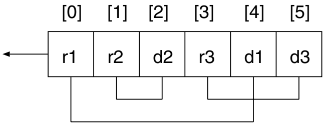


Fig. 1. Queue for delivery order

3 VRTPR-Tree

Most fundamental indexing structure for moving objects is called TPR-Tree [6][7] which adopts time-parameterized bounding rectangles called Conservative Bounding Rectangle (CBR). Self adjusting structure called Star-Tree was proposed in [10]. $R^{\exp}$ - Tree extended from TPR-Tree for expiration times of moving objects was proposed in [7]. TPR^{*}-Tree [11] employed a new set of insertion and deletion algorithms. We propose Vehicle Routing TPR-Tree (VRTPR-Tree) extended from TPR-Tree for moving vehicles. Although it has been applicable to moving objects in any dimension, we focus on moving vehicles in x - y dimensions in this paper. VRTPR-Tree is a height balanced tree associated with the feature of R-Tree [12]. We show the tree structure of VRTPR-Tree as follows.

3.1 Leaf Nodes

An entry E in a leaf node consists of a pointer to a vehicle and a time-parameterized bounding rectangle which bounds the vehicle as Equation (10). The form of the bounding rectangle depends on the position and the velocity vector of the vehicle and road network constraints. The time-parameterized bounding rectangle implies future position of the vehicle.

$$E = (v_j, BR(t)) \tag{10}$$

A bounding rectangle in a leaf node is defined as intervals on x and y coordinates as Equation (11).

$$BR(t) = ((BR_x^{\rightarrow}(t), BR_x^{\leftarrow}(t)), (BR_y^{\rightarrow}(t), BR_y^{\leftarrow}(t)) \tag{11}$$

Here, we define a new bounding rectangle called Routing CBR (RCBR). For simplicity, we consider only x coordinate as follows. The interval of RCBR at update time t_{upd} is equal to the position of bounded vehicle v_j as Equation (12).

$$RCBR_x^{\rightarrow}(t_{upd}) = RCBR_x^{\leftarrow}(t_{upd}) = \overline{v}_j(t_{upd}) \tag{12}$$

Moreover, we define a reachable rectangle $RR(t)$ of vehicle v_j within update interval I . Maximum moving distance of the vehicle within the update interval is calculated by Equation (13).

$$d_{\max j} = \vec{v}_j \times I \tag{13}$$

Let $P(t)$ be nodes which satisfy the condition $p \in P : d(\overline{v}_j(t), p) < d_{\max j}$. The reachable rectangle bounds the nodes $P(t)$ as Equation (14).

$$\begin{aligned} RR_x^{\rightarrow}(t) &= \min_P (px) \\ RR_x^{\leftarrow}(t) &= \max_P (px) \end{aligned} \tag{14}$$

The spread speed of the interval of RCBR is defined by using the reachable rectangle as Equation (15).

$$\begin{aligned} RCBR_{vx}^{\rightarrow} &= \min\left(vx_j(t_{upd}), - \frac{|\, RR_x^{\rightarrow}(t_{upd}) - x_i(t_{upd}) \,|}{I} \right) \\ RCBR_{vx}^{\leftarrow} &= \max\left(vx_j(t_{upd}), \frac{|\, RR_x^{\leftarrow}(t_{upd}) - x_i(t_{upd}) \,|}{I} \right) \end{aligned} \tag{15}$$

Hence, the interval of RCBR at time t is defined as Equation (16). RCBR spreads with the same speed as the vehicle and stops when it reaches at the reachable rectangle in the same direction; on the other hand, RCBR spreads until the reachable rectangle smoothly in the opposite direction of the vehicle.

$$\begin{aligned} RCBR_x^{\rightarrow}(t) &= \max\left(RR_x^{\rightarrow}, RCBR_x^{\rightarrow}(t_{upd}) + RCBR_{vx}^{\rightarrow}(t - t_{upd})\right) \\ RCBR_x^{\leftarrow}(t) &= \min\left(RR_x^{\leftarrow}, RCBR_x^{\leftarrow}(t_{upd}) + RCBR_{vx}^{\leftarrow}(t - t_{upd})\right) \end{aligned} \tag{16}$$

Accordingly, RCBR never spreads beyond the reachable rectangle and could bound vehicle even though the vehicle turns in a different direction. This improves the reliability and accuracy for prediction of future positions of vehicles.

Consequently, a bounding rectangle in leaf node is defined by using RCBR and the demands of the assigned customers (i.e., the riding nodes and dropping nodes) as Equation (17). Thus, the bounding rectangles imply not only the future position of vehicle but also responsible delivery area of the vehicle.

$$\begin{aligned} BR_x^{\rightarrow}(t) &= \min_i (rx_i, dx_i, RCBR_x^{\rightarrow}(t)) \\ BR_x^{\leftarrow}(t) &= \max_i (rx_i, dx_i, RCBR_x^{\leftarrow}(t)) \end{aligned} \tag{17}$$

3.2 Intermediate Nodes

An entry E in an intermediate node consists of pointers to child nodes and a bounding rectangle which always bounds bounding rectangles of the child nodes as Equation (18). Let M be the maximum number of entries in intermediate nodes to be fit in one node.

$$E = ((E_1, E_2, \cdots, E_m), BR(t)) : m < M \tag{18}$$

4 Assigning and Scheduling

Assigning customers to a vehicle depends on indexes of vehicles. Here, we define a demand rectangle of a customer as Equation (19). Nodes in VRTPR-Tree are evaluated by Equation (20) which is the responsible delivery area of vehicles in sub nodes, and a node with least value is selected from root to leaf. Finally, the customer is assigned to a vehicle which is pointed by selected leaf node. The least calculation order is $M \times \log_m(K)$, where M is the maximum number of entries in intermediate node, K is the number of vehicles, and $\log_m(K)$ is the least height of the tree.

$$\begin{aligned} DR_x^{\rightarrow} &= \min(rx, dx) \\ DR_x^{\leftarrow} &= \max(rx, dx) \end{aligned} \tag{19}$$

$$A(BR(t) \cup DR) \tag{20}$$

Assigned delivery demands (riding points and dropping points) are inserted into the queue of the vehicle we mentioned above in order of FIFO (first-in, first-out). More specifically, customers are transported one by one like taxis. This scheduling is unfit for share-ride buses, so that we propose a heuristic algorithm using λ-interchange for optimizing the initial solution in the next section.

5 Optimization Algorithm

Optimizing for assigning and scheduling is achieved by a heuristic approach using λ-interchange which is a basic algorithm for exchanging customers between vehicles. We consider two queues q_1 and q_2, and two sets of assigned customers C_1 and C_2 of vehicles v_1 and v_2 in Figure 2. Firstly, customers $C_1' \subset C_1$ of size $|C_1'| < \lambda$ are selected randomly, and the riding nodes and dropping nodes of C_1' is removed from q_1. Secondly, the riding nodes are inserted into random positions of

q_2, and the dropping nodes are inserted into the random positions on the right of the riding nodes. We define a series of exchange operators (λ_1, λ_2) where $|\lambda_1|, |\lambda_2| < \lambda$. For example, the operator $(1,0)$ indicates a shift of one customer from q_1 to q_2. The operator $(1,1)$ indicates an exchange of one customer between q_1 and q_2. We used $\lambda = 2$ so that all these operations have eight patterns. Figure 2 illustrates the operation $(1,0)$ on q_1 and q_2.

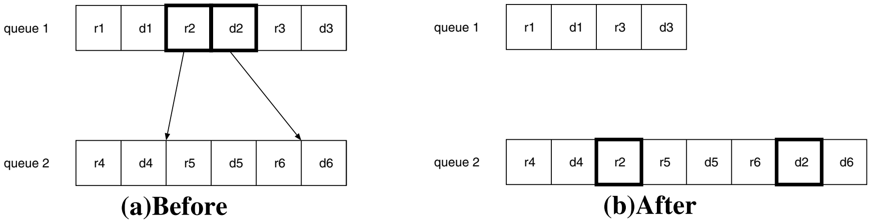


Fig. 2. Customer exchange between vehicles

For restricting optimizing area, we introduce height level of VRTPR-Tree. We define height *level1* as sets of leaf nodes included by bottom intermediate nodes. Height *level2* is sets of leaf nodes included by intermediate nodes above height *level1*. For example, in Figure 3, sets of three leaf nodes are height *level1*; a set of nine leaf nodes and three intermediate nodes is height *level2*. The λ-interchange applies to all of the combinations of vehicles in a set of leaf nodes. For example, in Figure 3, the number of combinations in the *level1* is ${}_3C_2 \times 3$; in *level2* is ${}_9C_2$.

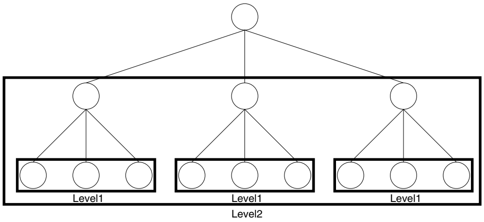


Fig. 3. Height level in VRTPR-Tree

After λ-interchange, a cost function is required for accepting or rejecting the solution. We define two cost functions based on traveling distance cost (TDC) or customer satisfaction cost (CSC). TDC is the sum of traveling times of two exchange vehicles as Equation (21). On the other hand, CSC is the inverse of the sum of customer satisfaction values of assigned customers to two exchange vehicles as Equation (22).

$$|q_i| + |q_j| \tag{21}$$

$$\frac{1}{\sum_{c_i \in C_i} CS_{c_i} + \sum_{c_j \in C_j} CS_{c_j}} \tag{22}$$

We used First-Best strategy which selects the first best solution with respect to the two cost functions. The heuristic process is repeated until a maximum time $T_{\max}$. Whether the obtained solution converges or not in the process is checked at fixed intervals ω.

6 Experiments

We used five experimental patterns shown in Table 1 with varying the number of customers from 100 to 200 on a machine with Pentium-4 processor. The parameter setting is as follows: road network is 21×21 grid (1000×1000 pixels), so that all intersections are orthogonal 4-crossed points except for four borders. Time limits of customers are set from between 1 and 5 times minimum riding time randomly. Other parameters are shown in Table 2.

Table 1. Experimental patterns

Pattern	Level	Cost
PT1	No optimization	
PT2	1	TDC
PT2	2	TDC
PT3	1	CSC
PT4	2	CSC
PT5		

Table 2. Parameter setting

Parameter	Value
$\lvert \vec{v} \rvert$	5
M	5
σ	0.5
$T_{\max}$	1000
ω	50

Averages of optimizing times are shown in Table 3. Optimizing times of *level 2* (PT3 and PT5) are about three times as much as optimizing times of *level 1* (PT2 and PT4), respectively. TDC is more quickly convergent cost than CSC.

Table 3. Averages of optimizing times

Pattern	Time(ms)
PT2	4607
PT3	10554
PT4	6112
PT5	18771

Figure 4 shows the traveling time needed to transport all customers, and Figure 5 shows the average of traveling distance of vehicles. Note that TDC produces good results; on the other hand, CSC produces bad results. The reason is, in an extreme case, that CSC tries to assign only a customer with low satisfaction to a vehicle, other customers with high satisfaction to other vehicle. Such case results in an increase of traveling time. Figure 6 and Figure 7 show the waiting time and the riding time of customers, respectively. The results indicate a similar tendency with Figure 4 and Figure 5. No optimization adopts FIFO queue, so that PT1 shows the lowest value in Figure 5. In Figure 7 which shows the average of customer satisfaction, CSC produces better results than TDC in contrast to other results. Obviously, there are trade-off relations between optimizing time and system performance and between traveling distance and customer satisfaction. We should choose a transportation pattern to suit customer needs and traffic circumstances.

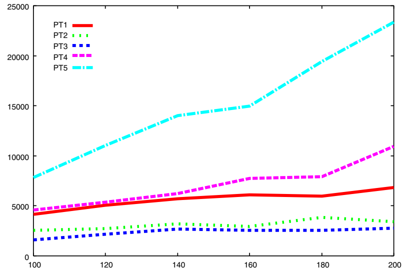


Fig. 4. Traveling time

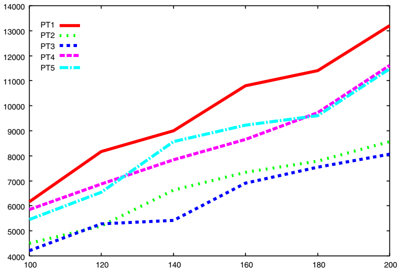


Fig. 5. Average of traveling distance

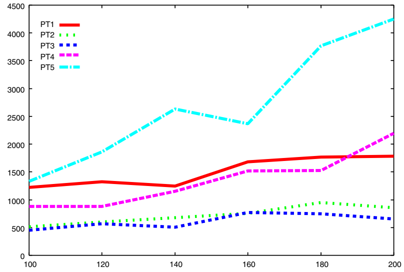


Fig. 6. Average of waiting time

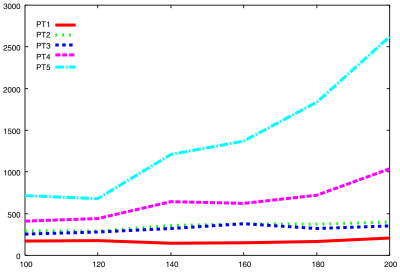


Fig. 7. Average of riding time

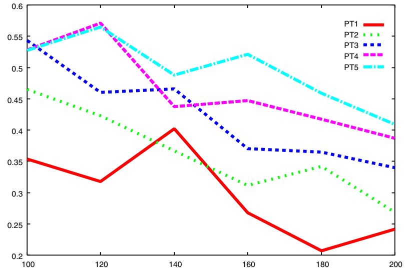


Fig. 8. Average of customer satisfaction

7 Conclusions

In this paper, we proposed VRTPR-Tree and a heuristic approach using λ-interchange to solve specific VRPTW. Firstly, VRTPR-Tree produces initial solution effectively in a short time. Secondly, a heuristic algorithm based on λ-interchange optimizes restricted parts of the initial solution. At the last, we performed experiments and compared five patterns. Our experimental results showed distinctive trends of the patterns. In our future works, we will apply other heuristics such as simulated annealing or genetic algorithm to our approach.

Acknowledgement. We would like to thank the 21st Century COE (Center of Excellence) Program for 2002, a project titled Intelligent Media (Speech and Images) Integration for Social Information Infrastructure, proposed by Nagoya University. And, we acknowledge to Prof. Naohiro Ishii of Aichi Institute of Technology for this perspective suggestion.

References

1. Thangiah, S.: Vehicle routing with time windows using genetic algorithms. Application Handbook of Genetic Algorithms, Vol. 2, (1995) 253-277
2. Potvin, J.Y, Bengio, S.: The vehicle routing problem with time windows – part2: Genetic search. INFORMS Journal on Computing 8, (1996) 165-172
3. Louis, S.J, Yin, X, M, Yuan, Z.Y.: Multiple vehicle routing with time windows using genetic algorithms. Proc. of the Congress on Evolutionary Computation, Vol. 3, (1999) 1804-1808
4. Ibaraki, T, Kubo, M, Masuda, T.: Effective local search algorithms for the vehicle routing problem with general time window constraints, Proc. of MIC, (2001) 293-297
5. M. Solomon.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, (1987) 35:254-264
6. Saltenis, S, Jensen, C.S.: Indexing the positions of continuously moving objects. Proc. Of ACM SIGMOD, (2000) 331-342
7. Saltenis, S, Jensen, C.S.: Indexing of moving objects for location-based services. Proc. Of ICDE, (2002) 463-473
8. Osman, I.H, N. Christofides.: Simulated Annealing and Descent Algorithms for Capacitated Clustering Problem. Research Report, Imperial College, University of London, (1989).
9. K.Q. Zhu, K.C. Tan, L.H. Lee, Heuristics for Vehicle Routing Problem with Time Windows, (1999)
10. Procopiuc, C, Agarwal, P, Har-Peled, S.: Star-tree: An efficient self-adjusting index for moving objects. Proc. Of ALENEX, (2002) 178-193
11. Tao, Y, Papadias, D, Sun, J.: The TPR*-tee: An optimized spatio-temporal access method for predictive queries. Proc. Of Very large data bases, (2003) 9-12
12. Guttman, A.: R-trees: a dynamic index structure for spatial searching. Proc. Of ACM SIGMOD, (1984), 47-57

Stochastic Learning Automata-Based Dynamic Algorithms for the Single Source Shortest Path Problem

S. Misra and B. John Oommen*

[1] School of Computer Science, Carleton University
Ottawa, K1S 5B6, Canada
{smisra, oommen}@scs.carleton.ca

Abstract. This paper presents the first Learning Automaton solution to the Dynamic Single Source Shortest Path Problem. It involves finding the shortest path in a single-source stochastic graph, where there are continuous probabilistically- based updates in edge-weights. The algorithm is a few orders of magnitude superior to the existing algorithms. It can be used to find the shortest path within the "statistical" average graph, which converges irrespective of whether there are new changes in edge-weights or not. On the other hand, the existing algorithms will fail to exhibit such a behavior and would recalculate the affected shortest paths after each weight change. The algorithm can be extremely useful in application domains including transportation, strategic planning, spatial database systems and networking.

Keywords: Learning Automata, Dynamic Shortest Paths, Intelligent Systems

1 Introduction

The problem of maintaining the shortest path information in a graph (with a single-source), where the edges are inserted/deleted and where the edge-weights constantly increase/decrease is referred to as the *Dynamic Single Source Shortest Path Problem* (DSSSP) [4,5,8,12]. The importance of the problem lies in the fact that it is representative of many practical situations in daily life. Our intention is to devise efficient solutions to maintain shortest paths while there are edge updates taking place on the structure of the graph. Out of the four possible edge-operations (insertion/deletion and increase/decrease), it can be shown that edge-insertion is equivalent to weight decrease, and edge-deletion is equivalent to weight-increase [4,5].

The well-known *static* solutions to the traditional combinatorial Single Source Shortest Path Problem [1,2] are unacceptably inefficient in such dynamic scenarios in practical life, because using them would involve re-computing the shortest path tree "from scratch" each time a topological change occurs in the graph.

Two of the earliest known works on the dynamic shortest path problem date back to the papers by Spira and Pan [13] and McQuillan *et al.* [7]. While the former is

* *Fellow of the IEEE*

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 239-248, 2004.

theoretically proven to be inefficient, the latter has neither been analyzed theoretically nor through simulations. The most recent and well-known solutions to the DSSSP on general graphs with positive real-valued edge-weights were proposed by Ramalingam and Reps [12], Franciosa *et al.* [3], and Frigioni *et al.* [5]. However, the solution by Franciosa *et al.* is limited only to the semi-dynamic problem. The currently acclaimed dynamic algorithms are constrained by the several limitations [14].

The work reported in this paper was inspired by the need of formulating an algorithm for finding the shortest path in such realistically occurring stochastic environments. The work seeks to find the shortest path for the "average" underlying graph (dictated by an "Oracle", also called the Environment). The purpose of this work is to find the "statistical" shortest path tree that will be stable regardless of continuously changing weights.

This paper presents a new algorithm that uses Learning Automata [6, 9] to generate superior results (when compared to the previous solutions). Learning is achieved by interacting with the Environment, and processing *its* responses according to the chosen actions. This is further clarified in Section 3 of the paper.

2 Previous Dynamic Algorithms

This section briefly discusses the two most significant solutions to the DSSSP problem. The detailed descriptions of the algorithms are more complex and are omitted here in the interest of brevity. They can be found in [5] and [12], and in the unabridged version of this paper [14].

2.1 Ramalingam and Reps' Algorithm (RR)

The first significant contribution to solving the fully DSSSP problem without recomputing everything "from scratch" was proposed by Ramalingam and Reps [12]. Their proposed algorithms were for cases of edge insertions/deletions, and are based on adaptations of the Dijkstra's solution to SP(G), the static version of the problem [2]. The authors of [12] showed that edge-insertions and deletions are equivalent to edge-length (weight) decrease and increase respectively. Inserting an edge can be abstracted to decreasing the edge-length from ∞ to a finite value, whereas decreasing an edge-length (weight) can be performed by inserting a new edge parallel to the edge under consideration. Similar arguments can be applied to deleting an edge or increasing an edge-length.

The *insertion* algorithm maintains a priority queue containing vertices with priorities equal to their distance from the endpoint of the inserted edge. When a vertex having a minimum priority is extracted from the priority queue, all the outgoing edges are traversed. The *deletion* algorithm works in two phases. The first phase of the algorithm determines those vertices and edges that are affected by the deletion of a particular edge, and removes those affected edges from $SP(G)$. The second phase determines the new output value for all the affected vertices and updates $SP(G)$.

2.2 Frigioni *et al.*'s Algorithm (FMN)

The second most significant solution to the fully DSSSP problem on digraphs with positive real weights was proposed by Frigioni *et al.* [4,5].

Briefly, the *weight-decrease* and *weight-increase* algorithms work as follows. The algorithms utilize the concept of the number of *output-updates,* which calculates the number of vertices affected by an input change in the graph. In the case of a weight increase, the number of output-updates is given by the number of vertices that change the distance from the source. If decreasing a weight decreases the distance of the terminating end of an inserted vertex from the source, a global priority queue is used to compute new distances from source. However, unlike the previous algorithms, on dequeuing a vertex, not all the edges leaving it are scanned. The "decrease" algorithm works in three phases; the first two are concerned with preprocessing, while the last one updates the distances of the vertices from the source, and is applied to the subgraph that is induced by the updated vertices.

The weight-increase algorithm is based on the following node-coloring scheme: (1) marking a node *white*, where such a node q changes neither the distance from s nor the parent in the tree rooted in s, (2) marking a node *red*, where such a node q increases the distance from s, and (3) marking a node *pink*, where such a node q preserves its distance from s, but it replaces the old parent in the tree rooted in s. There are three main phases of the algorithm. First, it updates local data-structures at the end-points of the affected edge, and checks whether any distances change. Then, the vertices are colored repeatedly by extracting vertices with minimum priority. Finally, the algorithm computes the new distances for the *red* vertices.

3 Learning Automata and the DSSSP Problem

3.1 Principles of Learning Automata

Learning Automata (LA) [6,9,15,16] have been used to model biological learning systems and to find the optimal action that is offered by a random environment. Learning is accomplished by actually interacting with the environment and processing its responses to the actions that are chosen, while gradually converging toward an ultimate goal. Learning Automata have found various applications in the past [10,12].

The learning loop involves two entities, the *Random Environ*ment (RE) and a *Learning Automaton.* Learning is achieved by the automaton interacting with the environment, and processing the responses it gets to various actions chosen. The intention is that the LA learns the optimal action offered by the environment. A complete study of the theory and applications of LA can be found in two excellent books by Lakshmivarahan [6] and by Narenda and Thathachar [9], and a recent issue of *the IEEE Transactions on Systems, Man and Cybernetics* [15], dedicated entirely to the study of LA.

The actual process of learning is represented as a set of interactions between the RE and the LA. The LA is offered a set of actions $\{\alpha_1, ..., \alpha_r\}$ by the RE it interacts with, and is limited to choosing only one of these actions at any given time. Once the LA decides on an action α_i , this action will serve as input to the RE. The RE will then respond to the input by either giving a *Reward*, signified by the value '0', or a *Penalty*, signified by the value '1', based on the *penalty probability* c_i associated with $\alpha_{i.}$ The LA learns the *optimal action* (that is, the action which has the minimum penalty probability), and eventually chooses this action more frequently than any other action. $\beta = \{0, 1\}$ is the set of inputs from the RE, where '0' represents a reward and '1' represents a penalty.

Variable Structure Stochastic Automata (VSSA) are usually completely defined in terms of *action probability updating schemes* which are either *continuous* (operate in the continuous space [0, 1]) or *discrete* (operate in steps in the [0, 1] space). The action probability vector P(n) of an r-action LA is $[p_1(n), ..., p_r(n)]^T$ where, $p_i(n)$ is the probability of choosing action α_i at time 'n', and satisfies $0 \leq p_i(n) \leq 1$, whose components sum to unity.

A VSSA can be formally defined as a quadruple (α, P, β, T), where α, P, β, are described above, and T is the updating scheme. It is a map from P × β to P, and defines the method of updating the action probabilities on receiving an input from the RE. Also they can either be ergodic or absorbing in their Markovian behavior.
An absorbing strategy is required. The updating rule that shall be used is analogous to the Linear Reward-Inaction (L_{RI}) scheme, well known in LA [9].

3.2 Motivation

As mentioned earlier, there is currently no solution to the DSSSP problem when the edge-weights are dynamically and stochastically changing. We shall attempt to extend the current models by encapsulating the problem within the setting of the field of LA with the three principal components of any LA system namely, the Automaton, the Environment, and the reward-penalty structure as described below.

The Automata. We propose to station a LA at every node in the graph. At every instance, its task is to choose a suitable edge from all the outgoing edges in that node. The intention, of course, is that it guesses that *this* edge belongs to the shortest path tree of the "average" overall graph. It accomplishes this by interacting with the Environment (described below). It first chooses an action from its prescribed set of actions. It then requests the Environment for the *current* random edge-weight for the edge it has chosen. The system computes the current shortest path by invoking either the RR or the FMN algorithms, whence the LA determines whether the choice it made should be rewarded or penalized as described below.

The Environment. The Environment consists of the overall dynamically changing graph. In the graph, there are multiple edge-weights which change continuously and stochastically. These changes are based on a distribution that is unknown to the LA, but assumed to be known to the Environment. In a religious LA-Environment feedback, the Environment also supplies a Reward/Penalty signal to the LA.

Reward/Penalty. Based on the action that the LA has chosen (namely, an outgoing-edge from a node which the LA *stochastically* "guesses" to belong to the shortest path tree), and the edge weight that the Environment provides, the updated shortest path tree is computed. The effect of *this* choice is now determined by comparing the cost with the current "average" shortest paths, and the LA thus infers whether the choice should be rewarded or penalized. The automaton then updates the action probabilities using an appropriate scheme, and the cycle continues. In this present paper, we have opted to use the L_{RI} scheme.

3.3 LA Solution to DSSSP: The LASPA Algorithm

The proposed LA solution to DSSSP, named as LASPA, is described below. There are two variants of LASPA: (i) LASPA-RR: when LASPA uses the algorithm proposed by Ramalingam and Reps [12], when an edge-weight increase/decrease occurs, and (ii) LASPA-FMN: when LASPA uses the algorithm proposed by Frigioni *et al.* [5]. Informally, the scheme is as follows:

1. Obtain a snapshot of the directed graph with each edge having a random weight. This edge-weight is based on the random call for an edge, where each edge-cost has its own mean and a variance. The algorithm maintains an action probability vector, P = $\{p_1(n), p_2(n) \ldots p_r(n)\}$, for each node of the graph.
2. Run Dijkstra's Algorithm to determine the shortest path edges on the graph's snapshot obtained in the first step. Based on this, update the action probability vector of each node such that the outgoing edge from a node, which is determined to belong to the shortest path edge, has an increased probability than before the update.
3. Randomly choose a node from the current graph. For that node, choose an edge based on the action probability vector. Request the edge-weight of this edge and recalculate the shortest path using either RR or FMN algorithms.
4. Update the action probability vectors for all the nodes such that the edges that belong to the shortest path's tree have a greater likelihood of being selected than before the update.
5. Repeat Steps 3-5 above until the algorithm has converged.

The formal pseudo-code of the algorithm is given below.

THE LASPA ALGORITHM

Input: (i) G(V,E) = A dynamically changing graph with simultaneous multiple stochastic edge updates occurring on it.
(ii) iters = total number of iterations
(iii) λ = learning parameter

Output: (i) A converged graph that has all the shortest path information
(ii) Values of all action probability vectors

```
BEGIN
G' = obtainAGraphInstance(G);   //Obtain a snapshot of the graph G
For (each vertex v in G')
   outdegree = checkNumOutgoingEdges(v);
   For (each outgoing edge e of v)
         InitialProbability(e) = 1.0/(outdegree);    //Initialize action probabilities
   End-For
End-For

updateActionProbabilityVector();  //Update action prob. vectors for all vertices
For (each vertex v from source s) //Consider all vertices and initialize structures
   executeDijkstraShortestPath(v); //Run Dijkstra's shortest path algorithm once
End-For

updateActionProbabilityVector();  //Update action prob. vector for all vertices
For (i = 0 to iters)   //Execute for all iterations
   randomVertex = getRandomVertex();  //Randomly choose a vertex.
   ap = getActionProbabilityVector(v); //Get action prob. vector for the vertex v
   e = chooseAnEdge (ap); //Choose an edge based on the action prob. vector for v
   currentWeight = getRandom(e); //Obtain a rand. val. of edge e from the env.
   oldWeight = getExistingWeight(e); //Obtain the existing weight
   If (oldWeight > currentWeight)
         executeDecreaseWeight(e); //Execute decrease weight algo. of FMN/RR
   End-If
   Else
         executeIncreaseWeight(e); // Execute increase weight algo. of FMN/RR
   End-Else
   updateActionProbabilityVector(); //Update action prob. vectors for all vertices
End-For
END
```

4 Experimental Details

Several experiments were designed to evaluate the performance of LASPA. Due to space limitations, this paper reports only some of the results obtained for the following sets of experiments:

(1) **Experiment Set 1:** Comparison of the performance of LASPA with FMN and RR for a fixed graph structure,

(2) **Experiment Set 2:** Comparison of the performance results with variation in graph structures, and

(3) **Experiment Set 3:** Sensitivity of the performance of LASPA to the variation of certain parameters, while keeping others constant.

Three *performance metrics* were used in the experiments: (1) *Average Number of scanned edges* per update operation, (2) *Average Number of processed nodes* per update operation, and (3) *Average time required* per update operation. It should be noted that the changes may occur randomly at any place in the network.

4.1 Experimental Results

This section reports only some of the results of all of the experiments that were conducted to examine the performance of LASPA with respect to the metrics described before. The results seem to show that LASPA doesn't perform well at the beginning, namely, when the algorithm is learning; but after the algorithm has learned, LASPA outperforms the RR and FMN algorithms. The results are summarized below.

Experiment Set 1

The implementations of RR, FMN, LASPA-RR, and LASPA-FMN were run on mixed sequences of 500 edge modifying update operations performed on a graph topology with 50 nodes and a 20% sparsity. The edge-weights were random real values having means between 1.0 and 5.0, and variances between 0.5 and 1.5. The value of the L_{RI} learning parameter was set to be 0.95.

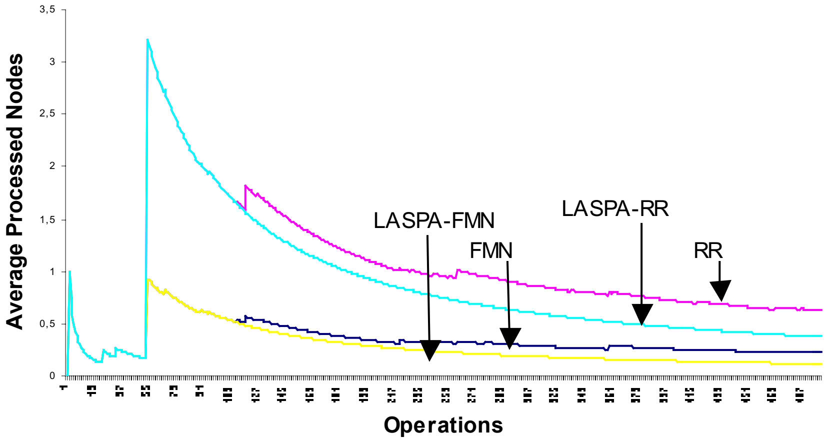


Fig 4.1. Graph showing the *average number of processed nodes* for the two algorithms FMN, RR and their LA versions, LASPA-FMN and LASPA-RR.

The results of the experiment for only average processed nodes and average time per update are shown in Figures 4.1 and 4.2 respectively. Initially, LASPA performs worse than FMN/RR. After convergence, the average number of nodes processed, the average number of edges scanned, and the average time spent per update operation are much less for LASPA when compared to both FMN and RR. This is evident from Figures 4.1 and 4.2. For example, in Figure 4.1, when the number of operations is 325, the average number of processed nodes for RR is around 0.9, whereas the average number of processed nodes for LASPA-RR is around 0.6.

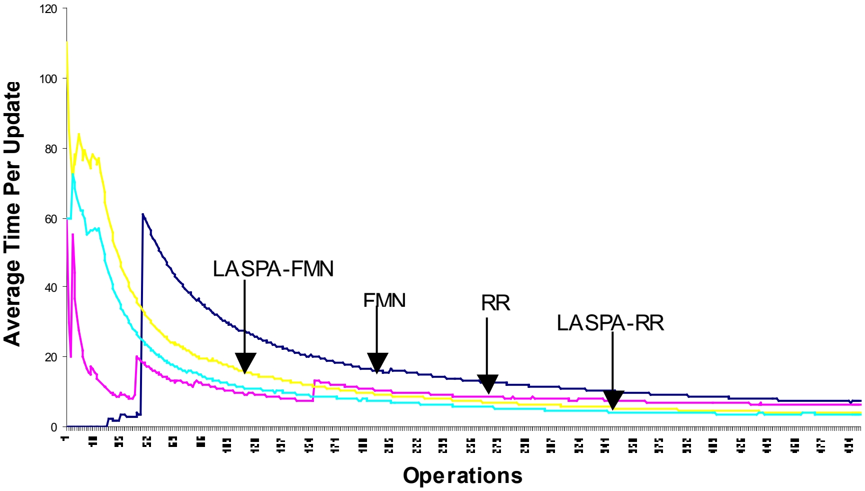


Fig 4.2. Graph showing the *average time per update* for the algorithms FMN and RR, and their LA versions, LASPA-FMN and LASPA-RR.

Experiment Set 2

A second set of experiments was conducted to evaluate the above results to investigate the results of the variation of graph structure (specifically: graph sparsity, and the number of nodes in graphs). In other words, we wanted to observe whether there was a different trend in the performance results when the structures of the graphs were varied, keeping other parameters constant. As shown in Table 4.1, the results obtained when the sparsity of graphs was varied, keeping other parameters constant.

A comparison of these values against each other demonstrates that both LASPA-FMN or LASPA-RR perform considerably better than the FMN and RR algorithms. For example, from Table 4.1, we see that the average value of time per update for LASPA-FMN and LASPA-RR at 10% sparsity are 3.12 and 3.66 respectively, whereas those of FMN and RR are 4.0 and 5.94 respectively. This shows that LASPA-FMN and LASPA-RR, on an average require less time per update operation than the FMN and RR algorithms. This is also true for the case when the number of nodes in the graph are varied.

Experiment Set 3

Figures 4.3 shows the sensitivity of the performance of the metric, the time per update operation, of LASPA-FMN to the variation in the learning parameter. Results for the metrics and with LASPA-RR yield similar results and are omitted here because e of space constraints. Figure 4.3 shows that as the value of the learning parameter increases, the average time per update also increases.

We have also conducted experiments to measure the sensitivity of the performance of LASPA to the increase in graph sparsity. Here too the performance is exactly as in the case of Figure 4.3 [14]. It can be observed that as the sparsity of the graph increases, the time for the update decreases. E.g, at 200th operation, the average time per update is around 6 ms when λ = 0.98, and around 4 ms when λ = 0.96.

Table 4.1. A table of the *Time Per Update* tabulated against the sparsity of the graphs The experiments performed with random graphs with mean edge costs between 1.0-5.0, variance between 0.5-0.9, λ=0.9, N=100, and number of operations=500.

Sparsity	FMN	RR	LASPA-FMN	LASPA-RR
10%	4.0/0.0/110.0	5.94/10.0/550.0	3.12/0.0/770.0	3.66/0.0/110.0
30%	3.16/0.0/110.0	7.4/0.0/2310.0	2.9/0.0/390.0	3.36/0.0/110.0
50%	1.92/0.0/60.0	2.76/0.0/280.0	1.76/0.0/550.0	2.78/0.0/160.0
70%	2.64/0.0/330.0	2.02/0.0/220.0	2.04/0.0/110.0	2.5/0.0/60.0
90%	1.1/0.0/60.0	1.23/0.0/110.0	1.06/0.0/60.0	1.56/0.0/60.0

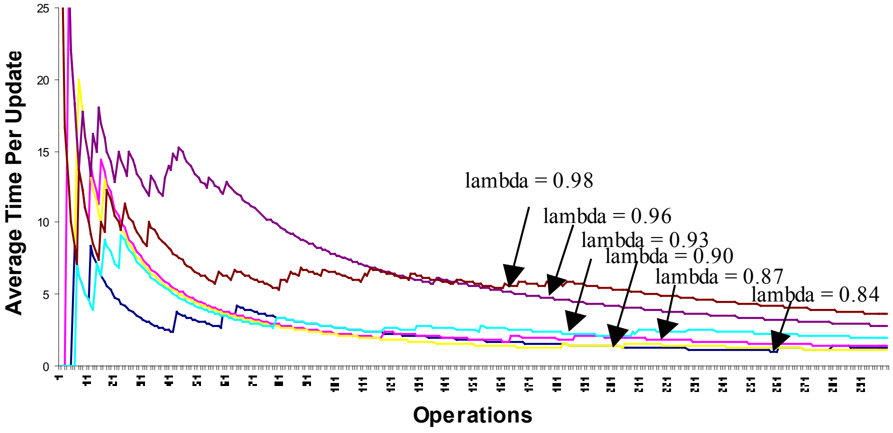


Fig. 4.3. Sensitivity of Average Time Per Update of LASPA-FMN to variation in λ.

5 Conclusions

This paper presents the first reported Learning Automata Solution to the Dynamic Single Source Shortest Path Problem. The proposed algorithm was implemented and rigorously experimentally compared to the two well-known fully-dynamic algorithms. The results show the superiority of the proposed Learning Automata-based algorithm. The *advantage* of the proposed algorithm is that in stochastic environments, it possesses a “statistical” shortest paths list that should be actual shortest paths irrespective of whether there are new changes in edge-weights taking place continuously. In such cases, the proposed solution converges to a shortest paths list, while the existing algorithms would recalculate affected shortest paths after every weight change.

References

1. Bellman, R.: On a Routing Problem. Quart. Appl. Math. Vol. 16 (1958), 87-90.
2. Dijkstra, E.W.: A Note on two Problems in Connection with Graphs. Numerische Mathematik. Vol. 1 (1959) 269-271.
3. Franciosa, P.G., Frigioni, D., Giaccio, R.: Semi-Dynamic Shortest Paths and Breadth First Search in Digraphs. STACS, Lect. Notes. Comp. Sc. Vol. 1200 (1997), 33-46.
4. Frigioni, D., Marchetti-Spaccamela, A., Nanni, U.: Fully Dynamic Output Bounded Single Source Shortest Path problem. ACM-SIAM Symp. Disc. Algo. (1996) 212-221.
5. Frigioni, D., Marchetti-Spaccamela, A., Nanni, U.: Fully Dynamic Algorithms for Maintaining Shortest Paths Trees. Journal of Algorithms. Vol. 34 (2000) 251-281.
6. Lakshmivarahan, S., Learning Algorithms Theory and Appls. Springer, New York(1981).
7. McQuillan, J., Richer, I., Rosen, E.: The New Routing Algorithm for the ARPANET. IEEE Transactions on Communications. Vol. COM-28, No. 5, (1980) 711-719.
8. Narvaez, P., Siu, K.-Y., Tzeng, H.Y.: New Dynamic Algorithms for Shortest Path Tree Computation, IEEE/ACM Transactions on Networking, Vol. 8, No. 6 (2000) 734-746.
9. Narendra, K. S., Thathachar, M. A. L., Learning Automata, Prentice-Hall (1989).
10. Oommen, B. J. and de St. Croix, E.V.: Graph Partitioning Using Learning Automata, IEEE Transactions on Computers, Vol. 45, No. 2 (1995) 195-208.
11. Oommen, B. J. and Roberts, T. D.: Continuous Learning Automata Solutions to the Capacity Assignment Problem, IEEE Trans. Comput., Vol. 49 (2000) 608-620.
12. Ramalingam, G. and Reps, T.: On the Computational Complexity of Dynamic Graph Problems, Theoretical Computer Science, Vol. 158, No. 1 (1996) pp. 233-277.
13. Spira, P. and Pan, A.: On Finding and Updating Spanning Trees and Shortest Paths, SIAM Journal of Computing, Vol. 4, No. 3 (1975) 375-380.
14. Misra, S. and Oommen, B.J.: The Dynamic Single Source Shortest Path Problem: Learning Automata Solutions. Unabridged version of the paper can be obtained from the authors as a technical report (Submitted for publication).
15. Obaidat, M.S., Papadimitriou, G.I. and Pomportsis, A.S.: Learning Automata: Theory, Paradigms and Applications, IEEE Trans. Syst., Man., and Cybern., Vol. 32, No. 6 (2002) 706-709.
16. Verbeeck, K. and Nowe, A.: Colonies of Learning Automta. IEEE Trans. Syst., Man., and Cybern., Vol. 32, No. 6 (2002) 772-780.

Multi-agent Based Integration Scheduling System under Supply Chain Management Environment

Hyung Rim Choi, Hyun Soo Kim, Byung Joo Park, and Yong Sung Park

Dong-A University, Department of Management Information Science,
604-714, 840 Hadan-dong, Saha-gu, Busan, Korea
{hrchoi, hskim, a967500, ys1126}@daunet.donga.ac.kr

Abstract. In order to satisfy customer's diverse demand and due date under SCM (Supply Chain Management) environment, this paper aims to establish effective scheduling in consideration of alternative machines and operation sequence of suppliers and outsourcing companies, and also focus on developing multi-agent based integration scheduling system to respond on a real-time basis to the various changes in the production environment. This paper has used genetic algorithm and multi-agent technology to develop this system. Compared with many other researches, this research has a great advantage in the sense that this multi-agent based integration scheduling system can reflect various changes in the production under SCM environment considering the situation of suppliers and outsourcing companies.

1 Introduction

In these days, as more attention is being paid to SCM (Supply Chain Management), the integrated scheduling under the SCM environment is preferred rather than the scheduling for one plant unit [3,5]. In particular, production cannot be performed by one producer alone. Rather, suppliers should provide raw materials in most cases, and also outsourcing companies should supply parts at the right time. Therefore, for effective and speedy production, we need integrated scheduling in consideration of the schedules of material suppliers and outsourcing companies. In addition, although the schedules of such individuals as suppliers, outsourcing companies, and producers are well optimized under the supply chain environment, there is a limit to the optimization of the entirety. Therefore, for the optimization of the entirety, producers need to establish an integrated scheduling considering the production plan of suppliers and outsourcing companies. As SCM is seeking the efficiency of entire value chain, the importance of scheduling for the optimization of the entirety cannot be exaggerated.

This integrated scheduling is closely related to the quick response to the diverse customer's demand, low inventory level, competitiveness improvement, and CTP (Capable To Promise). In addition to quality product, the strict observance of due date is also very important to customers. Therefore, the scheduling to decide a due date is a core part of supply chain management [5]. At present, researches on the integration of

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 249-263, 2004.

process planning and scheduling are being made for the production plan considering outsourcing companies and multi-plants. However, these researches also have a limit to the instant and real-time response to the diverse environmental changes under the SCM environment. The integrated scheduling under the SCM environment should be able to not only reflect the environmental changes of a producer, but also include the changes in the production environment of suppliers and outsourcing companies. For example, when one outsourcing company is out of operation owing to its machine failure, or the delivery of one supplier is delayed, the rescheduling should be made on a real-time basis. That is, integrated scheduling treats the suppliers and outsourcing companies as the divisions of one company. And if the producer cannot keep the due date for customers owing to environmental changes, he has to seek another supplier or outsourcing company to meet the due date.

To this end, this research makes use of a multi-agent system. A multi-agent system is emerging as a new paradigm to solve complicated problems under the diverse environmental changes [9]. A multi-agent system enables the solution of complicated problems under the diverse environment through communications between agents. Recently, many enterprises are moving towards open architectures for integrating their activities with those of their suppliers, customers and partners within supply chain. Agent-based technology provides a natural way to design and implement such environments. A number of researchers have attempted to apply agent technology to manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control. In MetaMorph II, several mechanisms were developed for dynamic scheduling and rescheduling by combining a bidding mechanism based on contract net protocol with a mediation mechanism based on the mediator architecture [10]. Also, Traditional approaches to planning and scheduling do not consider the constraints of both domains simultaneously. Agent-based approaches provide a possible way to integrate planning and scheduling activities through enterprise-level coordination between the product design system and the factory resource scheduling system. MetaMorph I implemented such a mechanism through enterprise level coordination between design mediators and resource mediators who in turn coordinate resource agents at the shop floor level [4,11]. But these studies are still not realistic because they had not practical methodology for integrating process planning and scheduling and dynamic scheduling. In this research, we design an effective integrated scheduling method considering the situational changes of suppliers and outsourcing companies, and at the same time try to propose multi-agent based integration scheduling system to respond to the various environmental changes under the SCM environment.

This paper consists of four chapters: the second chapter suggests the genetic algorithm-based integrated scheduling methodology; the third chapter deals with the structure and function of Multi-Agent system for integrated and dynamic scheduling, and also explains through a case study how this new system responds to the diverse environmental changes under the SCM environment. And finally, the fourth chapter deals with the contribution of this research.

2 Genetic Algorithm for Integrated and Dynamic Scheduling

This research uses genetic algorithm to establish integrated and dynamic scheduling. This algorithm considers alternative machines and operation sequence to integrate the process planning and scheduling, and also can perform rescheduling in response to the changes in the production environment. Traditionally, process planning and scheduling were achieved sequentially. However, this schedule didn't sometimes meet the realities of manufacturing spot, and also due to the bottleneck of certain resources, the scheduling couldn't be put into practice. Because of this, process planning and scheduling are integrated [1,6]. The integration of process planning and scheduling brings not only best effective use of production resources, but also practical process planning without frequent changes. Choi et al. [2] has proved that this integration of process planning and scheduling is far superior to the sequential process planning and scheduling in the aspect of due date.

Genetic algorithm considers the machine of outsourcing company to be alternative machine, and enables integrated scheduling considering alternative machines and operation sequence. Also, it enables rescheduling when changes have been made to the suppliers, outsourcing companies and producer. Here the objective function is the minimization of the makespan. In order to design genetic algorithm, first of all, the attribute of the problem should be analyzed, and then the presentation proper to the problem, performance measure, genetic operator, and genetic parameter should be decided. The following is genetic algorithm for the establishment of integrated scheduling considering alternative machines and operation sequence under the dynamic situation.

2.1 Genetic Algorithm for Integrating Process Planning and Scheduling

2.1.1 Representation

To achieve integrated production plan through genetic algorithm considering alternative machines and operation sequence, first of all, the problem should be represented in chromosome. The representation should be made in the way that all the processing sequence, alternative operation sequence and alternative machines could be decided. First, to represent processing sequence, the pattern to repeat the number of job as many as the number of operation is used. One gene means one operation, and in the represented order it will be allocated to the machines. For example, the problem of three jobs and three machines is represented in sequence as shown in the figure 1. The threefold repeated number in the first row is the number of job, and the reason that each job number has been repeated three times is that each job has three operations. The first repeat of job number means the first operation of the job, and the second repeat means the second job operation. If the job number continues to represent the number of job operation, this chromosome will always maintain its feasibility. The second row is the random numbers that will be used to decide alternative operation sequence. As each job is done in the one operation sequence, each job produces the same random number within the number of maximum alternative operation sequence. For example, as the job 2 in the table 1 has three alternative operation sequences, the random figure has to be produced within three. The third row has the random numbers to decide the alternative machine, producing them within the number of maximum alternative machines. In the table 1, the second operation of job 1 is to be done in the M2, but also can be done in the M1 and M3. In this case, the number of machine that can handle the second operation of the job 1 is 3. As there are no more alternative machines than this in the table 1, the random figures for all alternative machines will be produced within three. The index in the last row means the repeat number of job number, namely showing the ordinal operation of each job.

Table 1. Alternative machines and alternative operation sequences of each job

Job 1	**Operation**	**M1**	**M2**	**M3**
	sequence 1	**(M3)**	**(M1)**	
	(alternative machine)		**(M3)**	
	Operation	**M1**	**M3**	**M2**
	sequence 2	**(M3)**		**(M1)**
	(alternative machine)			**(M3)**
Job 2	**Operation**	**M1**	**M2**	**M3**
	sequence 1	**(M3)**		**(M1)**
	(alternative machine)			**(M2)**
	Operation	**M1**	**M3**	**M2**
	sequence 2	**(M3)**	**(M1)**	
	(alternative machine)		**(M2)**	
	Operation	**M3**	**M1**	**M2**
	sequence 2	**(M1)**	**(M3)**	
	(alternative machine)	**(M2)**		
Job 3	**Operation**	**M1**	**M3**	**M2**
	sequence 1	**(M3)**	**(M1)**	
	(alternative machine)		**(M2)**	
	Operation	**M1**	**M2**	**M3**
	sequence 2	**(M3)**		**(M1)**
	(alternative machine)			**(M2)**

3	2	2	2	3	1	1	1	3	
2	3	3	3	2	1	1	1	2	
3	2	2	1	1	2	1	1	2	
(1	*1*	*2*	*3*	*2*	*1*	*2*	*3*	*3)*	··· (Index)

Fig. 1. Chromosome representation

2.1.2 Selection Method

The seed selection is used as a way of selection [7]. Seed selection, as a way of individual selection that is used in the propagation of cattle and the preservation of individual, has been introduced to the evolution of genetic algorithm. If the random value of the individual, which belongs to the father among parents, comes within the figure of probability (0.9), the best individual will be selected within superior individuals from ranking population. But, if not, the individual will be randomly selected among the entire groups. The mother will be selected randomly among the entire groups, but in this case, first, two individuals will be selected randomly, and then the better individual based on the value of probability will be selected. These will be used as parents, and then returned to the individual groups, so that they will be used again later.

2.1.3 Genetic Operator

Crossover operator should maintain and evolve the good order relationship of chromosome. The crossover operator used in this research, first of all, produces a random section, and then insert all the genes inside the section into the parent 2. The position of insertion is just before the gene where the random section has started. If in the parents 1 the random section starts in the fourth place, then the position of insertion will be before the fourth gene in the parents 2. All genes of the random section are deleted with respect to their index of occurrence in the receiving chromosome. And then to make alternative operation sequence coincide to the same job number, the alternative operation sequence of the initial job number will be corrected. These processes will be performed alternating parents 1 and parents 2, thus producing two child individuals. After two offspring are evaluated, the better one will be sent as a next generation. The mutation operator gives a change to the chromosome, thus maintaining diversity within the group. This research uses the mutation operator based on the neighborhood searching method [7].

2.1.4 Objective Function and Replacement

The minimum makespan in the scheduling often means the highest efficiency of a machine. When a chromosome is represented as a permutation type, the makespan is produced by the process that assigns operations to the machines according to sequence of gene from left to right, while maintaining the technological order of jobs and considering its alternative operation sequence and alternative machine. The process is shown in figure 2. The next generation will be formed by the selection among current generation and with the help of genetic operator. The new individuals will be produced as many as the number of initial population, and form the next generation. By using elitism, bad individuals will be replaced with good individuals. Also, because of crossover rate and mutation rate, some individuals will be moved to the next generation without getting through genetic operator.

2.2 Genetic Algorithm for Dynamic Scheduling

The genetic algorithm suggested in this paper reflects the dynamic changes in the suppliers and outsourcing companies along with the production changes of the producer. When the production environment changes have happened in suppliers or outsourcing companies – the acceptance of new orders, machine failure, outage, and the absence of the worker in duty, all these changes will be reflected in the rescheduling. For example, when a machine cannot be operated for ten hours because of its failure, or a supplier cannot keep the lead-time, thus delaying 10 hours, the integrated scheduling will reflect the usable time of each machine and the possible starting time of each job. The figure 3 shows the process of rescheduling when a new order has been accepted at the time of t1.

(1) Chromosome

3	2	2	2	3	1	1	1	3
2	3	3	3	2	1	1	1	2
3	2	2	1	1	2	1	1	2

(2) processing sequence of each job

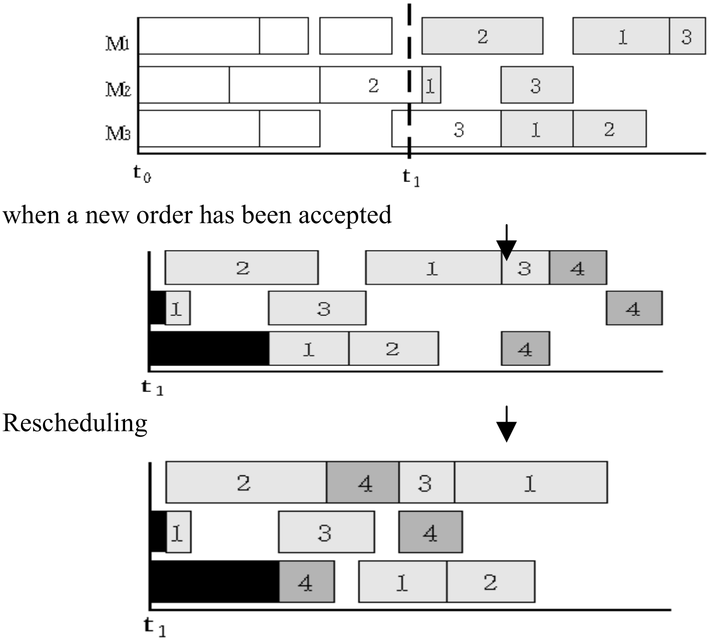
Job	Operation sequence (alternative machine)			
1	1	M1	M2	M3
		(M3)	(M1)	
			(M3)	
	2	M1	M3	M2
		(M3)		(M1)
				(M3)
2	1	M1	M2	M3
		(M3)		(M1)
				(M2)
	2	M1	M3	M2
		(M3)	(M1)	
			(M2)	
	3	M3	M1	M2
		(M1)	(M3)	
		(M2)		
3	1	M1	M3	M2
		(M3)	(M1)	
			(M2)	
	2	M1	M2	M3
		(M3)		(M1)
				(M2)

(3) processing sequence of each machine

Machine	sequence
M1	3 → 2
M2	2 → 3 → 1
M3	2 → 1

```
//"makespan"
// job_cnt : number of Job, machine_cnt : number of machine, genotype_length : length of chronomosome
for(int mm = 0 ; mm < job_cnt ; mm++)
{
 job_start[mm] = 0;
 job_next[mm] = 1;
}
 // job_start[mm] = ?;
 //different release time of a job

for(int kk = 0 ; kk < machine_cnt ; kk++)
{
 machine_start[kk] = 0;
 machine_next[kk] = 1;
}
 // machine_start[kk] = ?;
 //different release time of a machine

for(k = 0 ; k < genotype_length ; k++)
{
i = genotype_job_array[k];
s_1 = substitute_process[k];
s_2 = substitute_machine[k];

l = job[i][job_next[i]][s_1][s_2];
process_time = machine[l][machine_next[l]];

 start = 0;

 if(job_start[i] > machine_start[l])
 {
  start = job_start[i];
 }
 else
 {
  start = machine_start[l];
 }

 job_start[i] = start + process_time;
 machine_start[l] = start + process_time;

 ++job_next[i];
 ++machine_next[l];
}
```

Fig. 2. The evaluation of chromosome



when a new order has been accepted

Rescheduling

Fig. 3. Example of dynamic scheduling

In the process of rescheduling, the remaining jobs and the new jobs to be done by the new orders will be considered in the new production planning. It also considers the starting time of jobs and the usable time of machines. The starting time of jobs can be changed by the delay of supply or the operation delay of its prior process. The usable time of machines, as shown in the black shade of figure 3, can be changed by when the machine has already been allotted to other job, machine failure, and the absence of the worker in duty. All these dynamic changes will be reflected in the rescheduling [8].

3 Multi-agent System to Support Integration Scheduling

The Multi-Agent system is using genetic algorithm based integration scheduling methodology as a core engine. This system enables scheduling considering the capability of suppliers and outsourcing companies under the SCM environment. And based on the results of scheduling, proper suppliers and outsourcing companies can be selected. Accordingly, it enables speedy and effective response to the diverse environmental changes. In this respect, Multi-Agent system can lay the foundation for the optimization of the entirety under the SCM environment.

3.1 The Structure and Function of the Multi-agent System

As shown in the figure 4, Multi-Agent system is based on the structure of mediator. The system is composed of diverse agents. That is, registry server, order management, supplier management, outsourcing company management, inventory analysis, process planning, and scheduling. The central agent, "mediator" is coordinating all these sub-agents and also controlling the message exchange of them. This function of mediator can remove the bottleneck of message exchange between agents, while getting rid of the risk of malfunction. Under the central role of mediator, each agent exchanges communications, thus establishing integration scheduling considering the production environments of suppliers and outsourcing companies. Also, based on the integrated scheduling, the proper supplier and outsourcing company will be selected. The role and function of each agent in the Multi-Agent system are as follows:

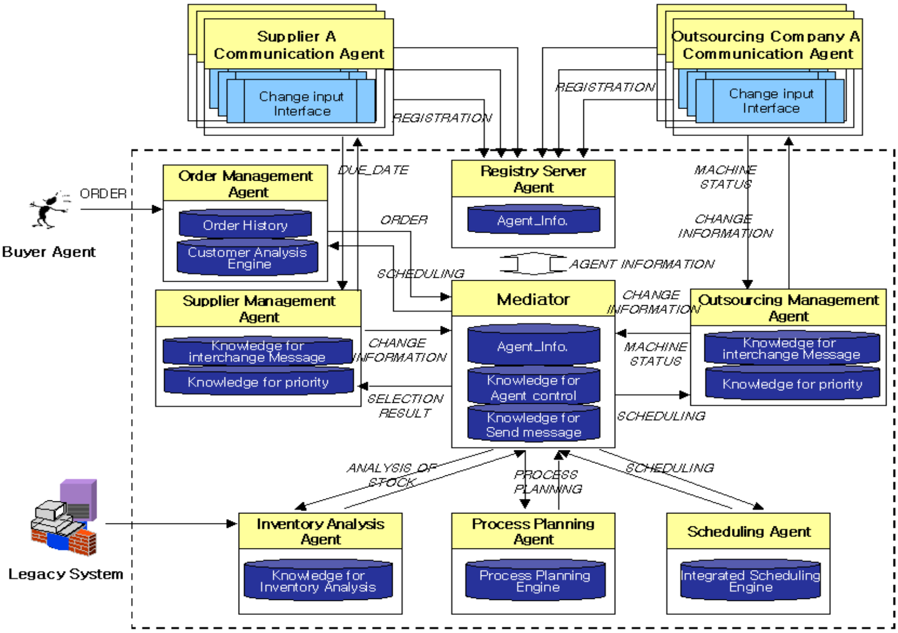


Fig. 4. The structure of the Multi-Agent system

1) Mediator.

Mediator plays the role of coordinating and controlling each agent and their message exchanges among themselves inside the system. To help many agents perform their diverse jobs, and transmit correct messages among agents, the central role of mediator is necessary. Mediator has information on agents and a knowledge base for the sake of controlling agents and message exchange.

2) Registry Server Agent (RSA).

A registry server agent has the agents of related suppliers and outsourcing companies registered. And these registered agents exchange communications among themselves. These registered companies are the main bodies of supply chain.

3) Order Management Agent (OrderMA).

An order management agent receives orders and confirms them. It keeps order information, and analyzes, classifies buyers through data mining and statistic analysis.

4) Supplier Management Agent (SMA).

A supplier management agent provides the information on the environmental changes in the suppliers to the inside system, and also transmits the information on supply schedule to the suppliers. This agent also keeps the suppliers priority on the basis their capability and confidence. This data will be used at the time of selecting suppliers when they all can keep the same due date.

5) Outsourcing Management Agent (OMA).

An outsourcing management agent provides the information on the machine situation of outsourcing companies for the sake of scheduling, and also, based on the scheduling, it makes a decision on the necessity of outsourcing. The criteria for the decision-making are whether the outsourcing company can keep the due date or not. When the producer cannot keep the due date for itself, this agent sends the message containing the necessity of outsourcing. The following is the agent's action knowledge represented in the form of IF-THEN, saying, "If the due date based on the production planning is not satisfactory, inform the mediator of the necessity of outsourcing." Here, Last_time is the finishing time of the last operation in the production planning.

```
IF  Last_time > due_date
Then send  yes_message  to  Mediator
```

This agent has the priority information on outsourcing companies, and this information will be used for selecting outsourcing companies.

6) Inventory Analysis Agent (IAA).

An inventory analysis agent analyzes the inventory level and makes a decision on the purchase of materials. The information on the inventory level is to be secured from the inside of the system. The inventory analysis agent makes a decision based on the purchase necessity analysis knowledge.

7) Process Planning Agent (PPA).

A process planning agent performs the role of process planning. This paper has used CBR (Case Based Reasoning) based process planning engine. The reason is that if the products of order-based producers are similar, the same process will be used. Choi et al. [3] has proved the availability of this methodology by applying it to molding industry.

8) Scheduling Agent (SA).
A scheduling agent performs the role of scheduling based on Genetic Algorithm based engine, considering alternative machines and operation sequence. This agent plays the critical role in the Multi-Agent system, and based on this scheduling, the supplier and outsourcing company will be selected.

9) Supplier and Outsourcing Company Communication Agent.
A supplier and outsourcing company communication agent performs the role of communications between Multi-Agent system and suppliers and outsourcing companies. For the establishment of scheduling, the supplier communication agent provides the possible due date of raw materials, and the outsourcing company communication agent provides the information on machine situation. These two agents provide Multi-Agent system with the information on the production environment changes through user interface.

3.2 The Integration Scheduling Process through Multi-agent System

The integration scheduling process of Multi-Agent system is composed of the followings: the process of scheduling for self-production, the process of scheduling for selecting outsourcing company, the process of scheduling for selecting supplier, and the process for rescheduling in case that the production environments of suppliers and outsourcing companies have been changed. If necessary, based on the rescheduling, the supplier and outsourcing company should be reselected.

1) The Process of Scheduling for Self-Production.
Figure 5 shows the case that a producer can make order-based products at his own factory without the help of suppliers and outsourcing companies. The producer establishes the scheduling for accepted orders, and provides the result to the buyer agent.

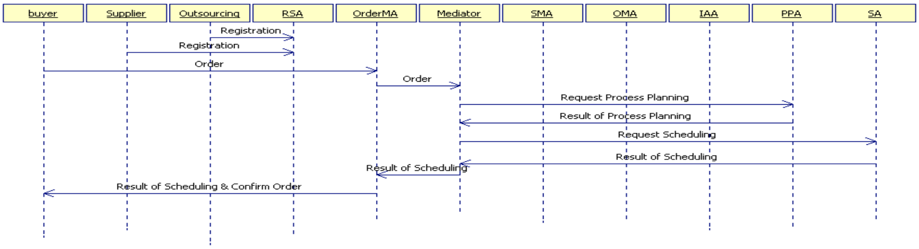


Fig. 5. The process of Scheduling for self-production

2) The Process of Scheduling for Selecting Outsourcing Company.
Figure 6 shows the case that as a producer cannot meet the required due date by self-production, he has to select an outsourcing company. The outsourcing management agent analyzes the necessity of outsourcing based on the scheduling, and if necessary, it asks for the information on the machine situation of outsourcing companies. Based on this information on machine situation, the scheduling agent establishes rescheduling. Based on this rescheduling and outsourcing company priority, the outsourcing management agent selects an outsourcing company.

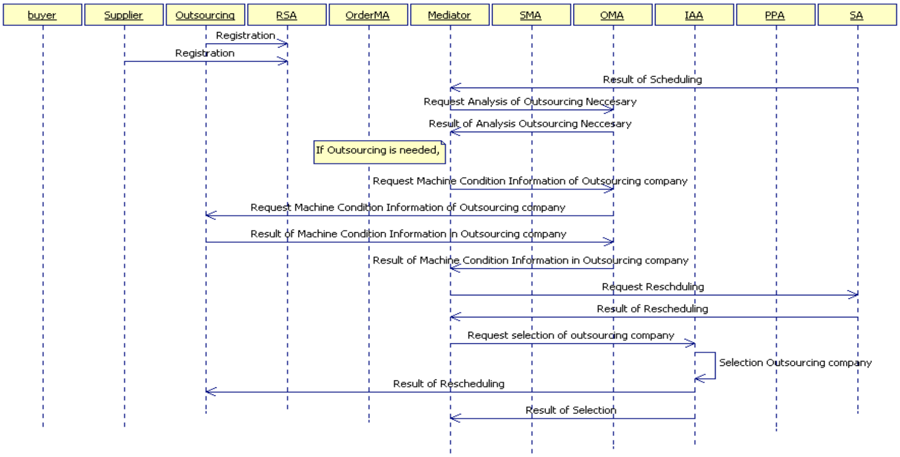

buyer
Supplier
Outsourcing
RSA
OrderMA
Mediator
SMA
OMA
IAA
PPA
SA
Registration
Registration
Result of Scheduling
Request Analysis of Outsourcing Neccesary
Result of Analysis Outsourcing Neccesary
If Outsourcing is needed,
Request Machine Condition Information of Outsourcing company
Request Machine Condition Information of Outsourcing company
Result of Machine Condition Information in Outsourcing company
Result of Machine Condition Information in Outsourcing company
Request Reschduling
Result of Rescheduling
Request selection of outsourcing company
Selection Outsourcing company
Result of Rescheduling
Result of Selection

Fig. 6. The process of scheduling for selecting outsourcing company

3) The Process of Scheduling for Selecting Supplier.
Figure 7 shows how to select the supplier in case that the producer doesn't have enough inventory of raw materials. The inventory analysis agent analyzes the inventory of the inside system and required raw materials for orders. If it thinks the supplier should be selected, the supplier management agent will ask the suppliers of possible due date, and based on this information, it will select a supplier.

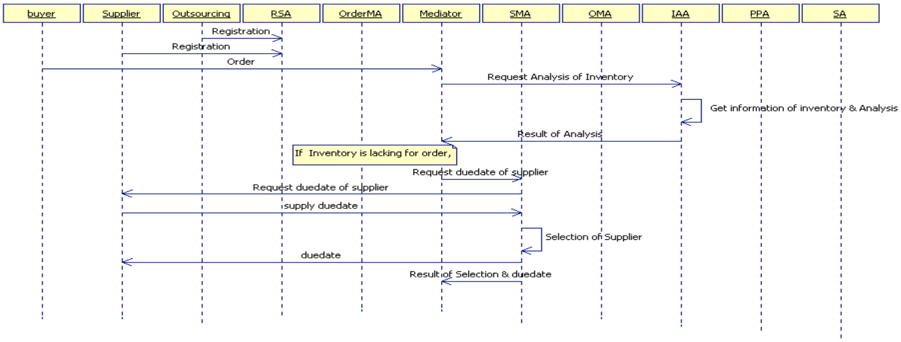

buyer
Supplier
Outsourcing
RSA
OrderMA
Mediator
SMA
OMA
IAA
PPA
SA
Registration
Registration
Order
Request Analysis of Inventory
Get information of inventory & Analysis
Result of Analysis
If Inventory is lacking for order,
Request duedate of supplier
Request duedate of supplier
supply duedate
Selection of Supplier
duedate
Result of Selection & duedate

Fig. 7. The process of scheduling for selecting supplier

4) The Process of Rescheduling to Production Environment Changes.
Figure 8 shows how to respond to the changes in the production environment. When there are changes in the production environments of outsourcing companies, the outsourcing communication agent provides this information to the outsourcing management agent. And based on this changed production environment, the scheduling agent achieves rescheduling, and the outsourcing management agent analyzes this rescheduling. However, if this rescheduling cannot meet the required due date, other outsourcing company should be selected.

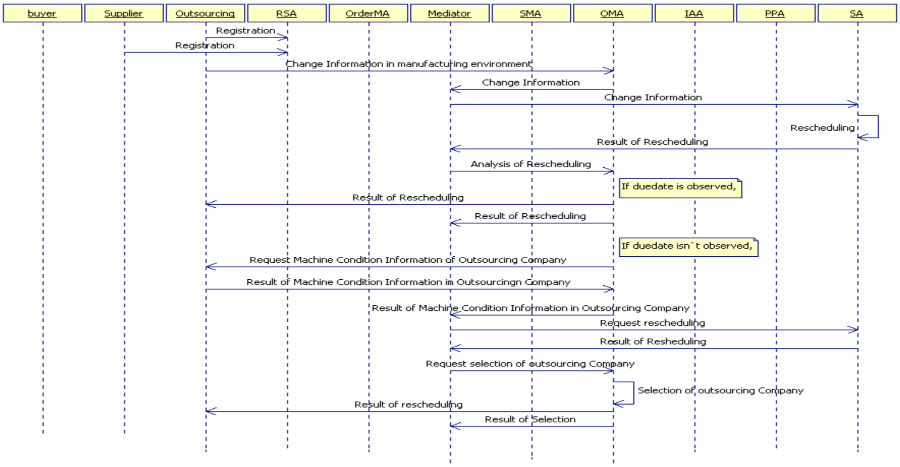

buyer
Supplier
Outsourcing
RSA
OrderMA
Mediator
SMA
OMA
IAA
PPA
SA
Registration
Registration
Change Information in manufacturing environment
Change Information
Change Information
Rescheduling
Result of Rescheduling
Analysis of Rescheduling
If duedate is observed,
Result of Rescheduling
Result of Rescheduling
If duedate isn`t observed,
Request Machine Condition Information of Outsourcing Company
Result of Machine Condition Information in Outsourcingn Company
Result of Machine Condition Information in Outsourcing Company
Request rescheduling
Result of Resheduling
Request selection of outsourcing Company
Selection of outsourcing Company
Result of rescheduling
Result of Selection

Fig. 8. The process of rescheduling to production environment changes

To testify the availability of the above processes of Multi-Agent system, this research adopts a molding company as a case study.

4 Case Study

In this research, we apply the Multi-Agent system to the real molding company. A molding company is a typical order-based producer. In particular, as the buyer has to produce their products by using this mold, the observation of due date is extremely important. Because of this, exact scheduling is badly needed. Anyway, Multi-Agent system can be applied to not only molding company, but also diverse order-based manufacturers. In the case study, “A” molding company transacts business with three suppliers and three outsourcing companies, and mainly outsources milling operation alone. First of all, it has developed the agents for three suppliers and three outsourcing companies and then has them registered in the registry server agent. The following is the process of integrated scheduling by step.

Step 1. Scheduling.
When “A” molding company accepts an order of molding product “cake-box,” which has a due date of 64 hours, it will start with the process planning and scheduling for this product. The alternative machine and alternative operation sequence that were not confirmed in the process planning will be included in the integrated scheduling, consequently establishing optimum process planning and scheduling. However, the due date from the integrated scheduling was 69 hours [2].

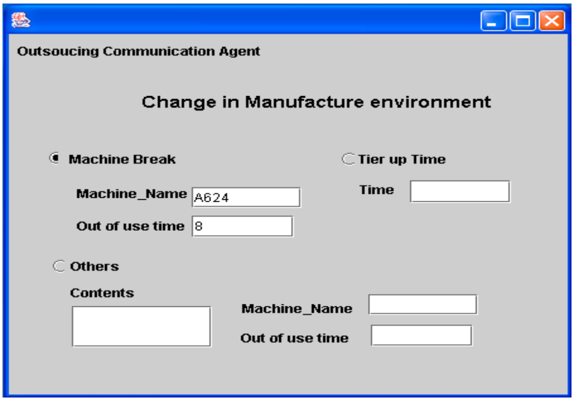
Step 2. Selection of Outsourcing Company.
As the makespan from integrated scheduling is 69 hours, it cannot meet the required due hour “64.” Therefore, the outsourcing management agent informs mediator of the necessity of outsourcing, and at the same time asks three outsourcing companies for their information on machine situation. Based on the information on machine situation, the scheduling agent will achieve rescheduling, and transmit its result to the outsourcing management agent through mediator. If the three outsourcing companies, all together, can meet the required due date, the outsourcing management agent will select one company based on the priority information of outsourcing companies.

Step 3. Selection of Supplier.
In order to produce “cake-box”, the mediator asks the inventory analysis agent to analyze the inventory level, and also make a decision on whether a molding company needs to buy raw materials or not. If its inventory runs short, the supplier management agent will ask three suppliers for their possible due date, and then compare this information with the scheduling received from the scheduling agent. The supplier management agent will select the supplier based on the scheduling and supplier priority information, and then transmit the due date.

Step 4. Responding to Production Environment Changes.
When the scheduled job is being delayed for 8 hours due to the machine failure during the process of production, this information will be inputted into the user interface as shown in the figure 9, and then the outsourcing communication agent will provide this information to the outsourcing management agent. The outsourcing management agent will provide the information to the scheduling agent through mediator.

The scheduling agent will achieve rescheduling considering the 8-hour failure of A624 machine. If the rescheduling cannot meet the due date, it will select the other outsourcing company.

On the other hand, if the supplier cannot meet the required due date, the supplier communication agent will transmit this information to the supplier management agent who will also inform the scheduling agent of this information through mediator. The scheduling agent is to achieve rescheduling based on the possible starting time of jobs. If the rescheduling can meet the due date, it will modify merely schedule information. Otherwise, it will select the other supplier company.



Outsoucing Communication Agent

Change in Manufacture environment

● Machine Break

Machine_Name A624

Out of use time 8

○ Tier up Time

Time

○ Others

Contents

Machine_Name

Out of use time

Fig. 9. The interface of outsourcing communication agent

5 Conclusion

In order to make production planning considering alternative machine and alternative operation sequence and also to respond on a real-time basis to the diverse production environment changes under the SCM environment, we proposed an architecture and methodology of Multi-Agent system for integration scheduling in this research. Former researches also developed integrated scheduling system considering outsourcing companies, but they couldn't reflect the diverse environmental changes on a real-time basis under the SCM environment. Under the SCM environment, it is very important to make integrated production planning considering outsourcing companies and suppliers, and simultaneously to respond to the diverse environmental changes on a real-time basis. In particular, under the production environment like an order-based producer where the observance of due date is a key factor, the importance of the integration scheduling system can never be exaggerated. The main implication of this paper is to propose an Multi-Agent system that can act of human experts who estimates the exact due date in a SCM by fully understanding the production capacity and scheduling to know whether the company can satisfy the due date of orders. The Multi-Agent system couples with optimization method to promptly schedule and select optimal partners as well as communication facilities in the dynamic SCM.

References

1. Brandimarte, P., Calderini, M.: A heuristic bicriterion approach to integrated process plan selection and job shop scheduling. International Journal of Production Research, 33 (1995) 161-181
2. Choi, H.R., Park, B.J., Park, Y.S., Kang, M.H.: Integrated Job Shop Scheduling considering Alternative Machines and Operation Sequence, Proceedings of Fall Conference of the Korean OR/MS society, (2003) 85-88

3. Lee, Y.H., Jeong, C.S., Moon, C.U.: Advanced planning and scheduling with outsourcing in manufacturing supply chain. Computers & Industrial Engineering, 43 (2002) 351-374
4. Maturana, F., Balasubramanian, S., Norrie, D.H.: A Multi-Agent Approach to Integrated Planning and Scheduling for Concurrent Engineering. In Proceedings of the International Conference on Concurrent Engineering – Research and Applications, Toronto. (1996)
5. Moon, C.U., Jeong, C.S., Lee, Y.H., Kim, J.S.: GA based Heuristic for Integrated Process Planning and Scheduling in Supply Chain, Proceedings of International Conference on Production Research, Bangkok, (2000)
6. Palmer, G.J.: A simulated annealing approach to integrated production scheduling, Journal of Intelligent Manufacturing, 7 (1996) 163-176
7. Park, B.J., Choi, H.R., Kim, H.S.: A Hybrid Genetic Algorithms for Job Shop Scheduling Problems, Genetic and Evolutionary Computation Conference Late-Breaking Papers, July 7-11, E. Goodman, ed., ISGEC Press, San Francisco, (2001) 317-324
8. Park, B.J.: A Development of Hybrid Genetic Algorithms for Scheduling of Static and Dynamic Job Shop, Ph.D. thesis, Department of Industrial Engineering, Dong-A University, (1999)
9. Shu, S., Norrie, D.H.: Patterns for Adaptive Multi-Agent Systems in Intelligent Manufacturing, Proc. of the 2nd International Workshop on Intelligent Manufacturing Systems, Leuven, Belgium, (1999) 67-74
10. Shen, W., Norrie, D.H.: An Agent-Based Approach for Dynamic Manufacturing Scheduling, In working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN.
11. Shen, W., Norrie, D.H.: Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey, Knowledge and Information Systems, 1, 2 (1999) 129-156

A Representation of Temporal Aspects in Knowledge Based Systems Modelling: A Monitoring Example

Jose A. Maestro, César Llamas, and Carlos J. Alonso

Dpto. de Informática, Universidad de Valladolid (Spain)
{jose, cllamas, calonso}@infor.uva.es

Abstract. A most common requirement in the development of Knowledge Based Systems in dynamic environments is the capability of expressing time. This paper presents how it is possible to express time related requirements on KBS tasks and to include time explicitly in rules. Such kind of facilities is attained using UML diagrams embedded in the usual CommonKADS notation preserving the methodology. A continuous monitoring task is used to illustrate thorough the paper. Some specific lacks, in CommonKADS, for the accurate analysis of these tasks in the time domain are identified, and the corresponding adaptations are presented. Finally, in order to express temporal elicited knowledge, a notation to include time in rules, focused on instants and intervals, is added.

Keywords: KBS Methodology, CommonKADS, real-time systems, monitoring.

1 Introduction

The ability of expressing time in Knowledge Based Systems (KBS) is an essential feature when treating with applications devised to monitoring and diagnosing of continuous processes in manufacturing plants. Actually, several applications with these requirements have been developed in our workgroup to be applied in a beet sugar factory (Aerolid, Turbolid, Teknolid) [1,2,3,4]. In order to implement such kind of systems, a real time expert systems platform like G2© (Gensym) has been employed. This toolkit offers facilities to express knowledge in the form of frames, rules, procedures and so on. Also, as it is expected, this tool permits dealing explicitly with time, timed parameters and time triggered rules.

However, usual knowledge based methodologies do not include specifications for dynamic behaviors. Hence, these methodologies only allow obtaining a static description of the system. For instance, the most successful European methodology for KBS, CommonKADS, being useful and widely known and used, does not offer the adequate facilities required in order to the problem solving methods be expressed in a temporal situation. It is not our opinion that, CommonKADS must change substantially to be a usable methodology in Soft Real-Time Knowledge Based Systems (Soft RT-KBS) and some attempts have just been made to adapt the methodology to be able to deal with Hard RT-KBS [5,6].

Nowadays, it seems to be that UML should be an integral part of every modern specification. The original KADS [12] description techniques have been partially

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 264-273, 2004.

changed for UML diagrams; the textual task specification can be substituted by activity diagrams for instance, and a general Object-Oriented approach has been chosen in the CommonKADS methodology [10]. Therefore, special attention to software engineering methodologies that try to integrate UML and RT Systems [7,8,9] must be taken, and especially to the new UML Profile for Schedulability, Performance and Time Specification [11] should be taken into account.

To acquire a general overview of the proposal, an example is developed throughout the paper. The example consist of the analysis of a continuous monitoring task, developed using the proposed techniques and notations. In our domain the task relies on a set temporal relations, and includes tasks that must be scheduled periodically. In its analysis, CommonKADS, some extensions for real-time [6] and UML are employed. The task is a model driven monitoring and the problem is expressed in terms not related with temporal reasoning, although some notation to include time in the rules is added. Moreover, the problems concerning the real time platform requirements and schedulability are not taken into account.

In the next section, a first approximation to the monitoring task is considered, in the framework described in [12,13,10]. In Section 3 the specification of the task in terms of CommonKADS augmented with the additional notation required is introduced. In the Section 4, a simple notation adequate for expressing time into rule is presented and employed into a rule type generalized to be used in the monitoring task. The conclusions and an additional example finish.

2 Monitoring a Continuous Environment

The monitoring task plays an important role in diagnosis and supervision systems [14]. Even more, it has a special status in the OLID generation of automated supervisory systems [4]. The main responsibility of a monitoring task is: observe the system evolution, in order to determine whether exists an abnormal behavior; Fig. 1 shows the monitoring task location, as an isolated module previous to visualization and diagnosis modules.

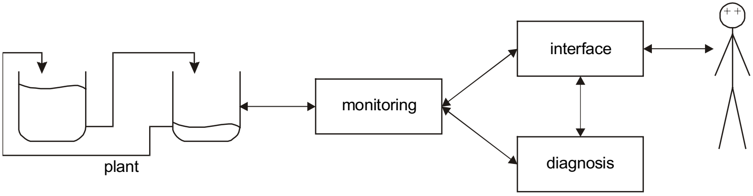


Fig. 1. Location of the monitoring module in the application system.

The monitoring task, as is presented in Alonso *et al.* [3,15] could be described as in Fig. 2. This task comprises two subtasks: *normal monitoring* and *intensive monitoring*, being, the former, a simple model driven monitoring, like the one mentioned by Breuker *et al.* in [13] and Schreiber *et al.* in [10]. The latter is a more specific model driven monitoring task, more time consuming, and subtle to be triggered to confirm the existence of a discrepancy (*see* Alonso *et al.* [3]).

Furthermore, each subtask have a specific execution period. The overall monitoring process integrates both modes; Fig. 2 illustrates the monitoring task as can be seen by the rest of the system.

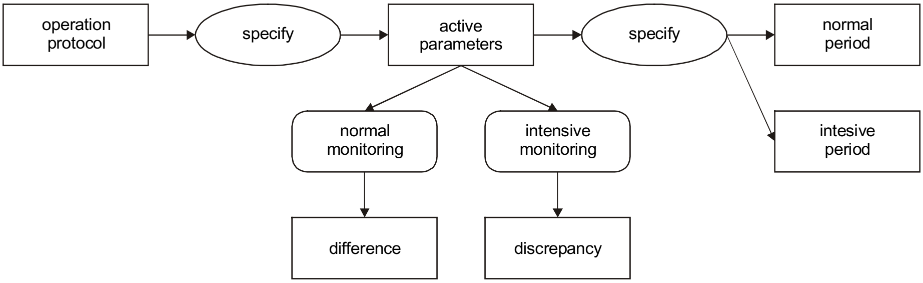


Fig. 2. High level inference structure for the *monitoring* task.

The notion of trajectory deviation considered is described in terms of knowledge representation involving the role *difference* and temporal constraints. Rules in both tasks, *normal* and *intensive* monitoring, contain references to thresholds and time deadlines on each parameter.

Either of these tasks is scheduled in a regular basis and its behavior is described appropriately in terms of time intervals. Several situations could change the steady execution state of them: (*i*) a change of operation protocol, that convey a transient situation that require the monitoring to be suspended, and possibly a different schedule rate when the normal operation is resumed, and (*ii*) a change of task scheduling from *normal* to *intensive* monitoring, and *vice versa*.

The monitoring, described in [3], relies on the concept of *monitored variable* that will be employed in the domain layer. A monitored variable could present three states: *normal*, *vigilance* and *critical*; its current state depends on roles *difference* and *discrepancy* and some temporal considerations.

The *intensive monitoring* begins to be scheduled in case of the monitoring variable gets a state different from the *normal* one. The effective distinction between *normal* and *intensive* monitoring makes possible to operate at different rates under each operation protocol. Intensive monitoring needs a higher scheduling rate; hence, in this way it is possible to invoke this task only when it is strictly necessary [15]. In addition, this task behaves in a way rather different from normal monitoring. Intensive monitoring has a different set of relations from the normal monitoring and interleaves, in some sense, with the diagnosis subsystem [1].

In Alonso *et al.* [3] a knowledge level description for the supervision and diagnosis, using KADS [13], is presented. The monitoring model appears embedded, partially, within the *supervision and diagnosis* task. In this description, the independence of the tasks from the specific elements of the application environment has been preserved (*e. g.* from time constraint). However, the representation obtained is far from being satisfactory, and presents some drawbacks: (*i*) the flow control is very complex, (*ii*) some constructors disrupt the sequential execution of the task (`fork`, `wait_until` and `break_if`), and (*iii*) the inference structure becomes complex, possibly as a consequence of the previous reasons.

3 Analysis of the Continuous Monitoring Tasks

In a non-static domain, most of the knowledge involved has a dynamic flavor. There will exists some timing restrictions and dynamic and temporal relations, which are elicited knowledge and, therefore, must be included in a suitable manner in the knowledge model. The task description for the monitoring could be enhanced noticeably if the schedule, for the normal and intensive monitoring of the set of parameters, could be expressed appropriately. Also, a more adequate representation for the control structure is required – in this case, an activity diagram in UML is more descriptive and easier to understand than a algorithmic textual description.

```
  TASK monitoring ;
    GOAL : "Analyze an ongoing process to find an abnormal
      behaviour";
    ROLES :
      INPUT :
        Protocol : "Current system operation point";
      OUTPUT :
        discrepancy : "Indication of deviant system
          behavior";
    SPECIFICATION : "Watch the system evolution to discover
      whether any parameter behaves not according to system
      expectations" ;
  END TASK

  TASK-METHOD two-step-monitoring ;
    REALIZES : monitoring ;
    DECOMPOSITION :
      INFERENCES :
        specify ;
      TASKS :
        normal-monitoring, intensive-monitoring ;
    ROLES :
      INTERMEDIATE :
        active-parameters : "Set of parameters to observe
          the system evolution";
       parameter : "A parameter to be monitored" ;
        normal-period : "Period to monitor a parameter
          showing normal behavior";
        intensive-period : "Period to monitor a parameter
          showing abnormal behavior";
    CONTROL STRUCTURE :
      specify (protocol \to active-parameters) ;
      FOR-EACH parameter IN active-parameters DO
        Specify (parameter → normal-period +
                                         intensive-period) ;
        ACTIVITIES one-parameter-monitoring DO
          normal-monitoring (normal-period, parameter →
                                               difference) ;
          intensive-monitoring (intensive-period,
                                 parameter → discrepancy) ;
        END ACTIVITIES
      END FOR-EACH
END TASK-METHOD
```

Fig. 3. Task and method description for the *monitoring* task.

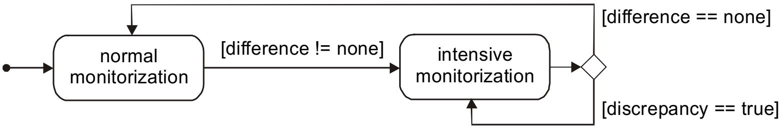


Fig. 4. Activity diagram for *one-parameter-monitoring*.

In practice, good methodology for use in real-time systems design would contemplate the use of diagrams and other textual ways of expressing the temporization of the tasks, events and state changes in the system [16]. In the Fig. 3, a task method specification for our monitoring task is presented. It employs a new primitive, `ACTIVITIES`, bounded to the activity diagram *one-parameter-monitoring* that appears in Fig. 4. This diagram permits us to represent the activity states of the monitoring task, without tangling the control structure of the monitoring task.

```
TASK normal-monitoring ;
  GOAL : "Analyze an ongoing process to find an abnormal
    behaviour";
  TYPE : periodic ;
  RELATIVE-TIME : YES ;
  PERIOD : period ;
  ROLES :
    INPUT :
      parameter: "A signal which behaviour is being
        analyzed" ;
      period : "Period which the task is realized" ;
    OUTPUT :
      difference : "Indication of deviant parameter
        behavior" ;
  SPECIFICATION : "Watch the parameter evolution to
    discover whether it behaves not according to system
    expectations" ;
END TASK

TASK-METHOD system-driven-monitoring ;
  REALIZES : normal-monitoring ;
  DECOMPOSITION :
    INFERENCES : compare, specify ;
    TRANSFER-FUNCTIONS : obtain ;
  ROLES :
    INTERMEDIATE :
      norm : "Expected normal value for the parameter" ;
     parameter-value : "Current value for the
       parameter" ;
  CONTROL STRUCTURE :
    specify (parameter → norm) ;
    obtain (parameter → parameter-value) ;
    compare (parameter-value + norm → difference) ;
END TASK-METHOD
```

Fig. 5. Task description for the *normal monitoring.*

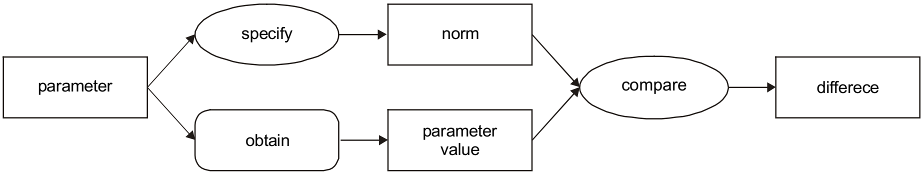


Fig. 6. Inference structure for the *normal monitoring* method.

Timing constraints can be expressed with little changes in the standard CommonKADS task description as it is displayed in Fig. 5 (see also Fig. 9 in the appendix). Tree additional fields –`type, period, relative-time`– have been added to the standard notation in the task description, following the ideas proposed by Henao *et al.* in [5, 6].

4 Rule Time Notation

The special nature of the time knowledge relations makes difficult to grasp them into the usual CommonKADS rule notation. In our domain, there are temporal relations that must be expressed in the knowledge base – for example, in Fig. 7 a complex state diagram is presented, which includes conditional and temporal transitions. Maybe, the usual state diagram is the most powerful tool for describing the knowledge underlying this kind of systems [7], being intuitive and permitting being formalist.

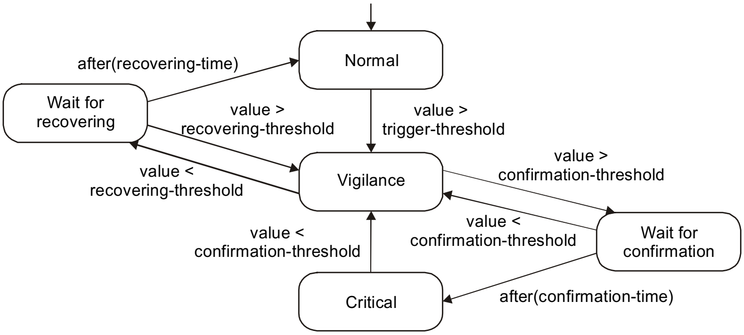

Normal
after(recovering-time)
Wait for recovering
value > recovering-threshold
value > trigger-threshold
value < recovering-threshold
Vigilance
value > confirmation-threshold
value < confirmation-threshold
value < confirmation-threshold
Wait for confirmation
Critical
after(confirmation-time)

Fig. 7. State diagram for a monitored variable.

This state diagram represents all the necessary knowledge to identify a discrepancy and to recover from it. As can be seen, a monitored variable has three main states and two auxiliary waiting states, associated to the intervals processing. However, as CommonKADS stands a textual description for rules rather than a graphical one, it is necessary to translate the temporal transitions into the standard rule notation.

To describe the rule content a kind of zero order logic alike notation is proposed by Schreiber *et al.* in [10]. Despite the fact that CommonKADS encourages the use of this plain language, it is also assumed that sometimes it is necessary a first order logic notation to describe the rules. Whilst CommonKADS includes an almost entire BNF notation definition of CML2 (Concept Modelling Language), it does not include a specific definition for rule content (*see* [10, ch. 14]). It could be used as a way to extend the methodology to those kinds of problems that are not taken into account, such as dynamic domains.

In Fig. 8 a rule-type and abstract rule instances are shown. These rules reflect the state diagram shown in the Fig. 7 into a CommonKADS rule description alike. A kind of temporal first order logic has been employed. This logic is inspired in the work of *Allen and others* compiled by Galton in [17]. Also, these rules are straightforward to be coded into a temporal expert system shell such as Gensym G2 ©.

Monitoring has only references to past and present time so that a linear time model is proposed. Instants and intervals are necessary to express our monitoring model. The usual `FORALL` quantifier has been replaced with a more expressive `ALWAYS`. This is a temporal quantifier so that an interval must be provided. In the same way it can be included another quantifier, `SOMETIME`, together with the set of classical *Allen's* predicates to deal with intervals and instants. Additionally, it is supposed each monitored variable has an historical that maintains past data.

The approach shown is a generalization of the one chosen in the OLID generation of industrial supervisory systems [1,2,3,4]. The supervision was made over an medium-size industrial plant with hundred of control loops. In these cases, monitored variables represent physical magnitudes like pressures, levels, flows or temperatures. To ensure the generality and usefulness of the monitoring task, each signal was divided into two monitored variables: HIGH and LOW ones. To avoid a massive off in variables, this three threshold technique was intended. It allows to filter spurious data and to ensure that a *critical* state value points out a potential problem.

```
RULE-TYPE monitoring-rule ;
  ANTECEDENT: monitored-variable ;
    CARDINALITY: 1 ;
  CONSEQUENT: monitored-variable ;
    CARDINALITY: 1 ;
  CONNECTION SYMBOL: detect ;
END RULE-TYPE

MV.state = normal AND MV.value > MV.trigger-threshold
  DETECT
MV.state = vigilance

MV.state = vigilance AND
                 ALWAYS t IN [NOW - MV.i-confirmation, NOW]
                     MV.value(t) > MV.confirmation-threshold
  DETECT
MV.state = critical

MV.state = critical AND
                        MV.value < MV.confirmation-threshold
  DETECT
MV.state = vigilance

MV.state = vigilance AND
                   ALWAYS t IN [NOW - MV.i-recovering, NOW]
                      MV.value(t) > MV.recovering-threshold
  DETECT
MV.state = normal
```

Fig. 8. The rule set for the monitoring tasks. MV stands for "monitored variable".

The complete supervisory system comprises other tasks such as diagnosis, human-machine interface, or data validation. The monitored variable structure briefly described, not only let to perform detection but also helps to perform fault diagnosis. They can be used to narrow the search if a cause-effect tree, including diagnosis information directly related for one variable, is associated to each monitored variable as it is described in [3, 18].

5 Conclusions

A model for a real-time task of a knowledge-based system has been presented. In this model CommonKADS and UML have been employed in order to represent suitably the knowledge involved and the dynamic behavior of the task. This integration has been made following Schreiber (*et al.*) latest CommonKADS [10] methodology book recommendations, and a *monitoring* task, that is a generalization of those used in the OLID generation, has been described.

It has been made clear that, if the system to be analyzed has a temporal description, the usual notation to describe the task method could not be the best choice. Furthermore, Real-Time systems demands its own concepts and notation, usually a graphical one, which is difficult to describe using pseudo code without yielding to a procedural program.

Hence, an alternative to the method specification of the monitoring task has been used, in this case, the activity diagram. This alternative, founded in the UML set of diagrams, has just been justified in the framework of the CommonKADS methodology. This kind of diagram allows us to describe sequences of tasks and the possible parallelism among them without having to write constructions as the usual `fork-join` couple.

State diagrams have been employed in wherever situations the expressivity could be enhanced. In this paper, apart from the activity diagram, a state diagram has been presented for the monitored variable. This diagram, also, solves partially the problem of expressing time relations in contrast to the pseudo code form.

The rule notation has been augmented to express time dependencies. This has permitted to simplify the task description. Despite the fact that the time model chosen is simple, it can cover a broad range of applications.

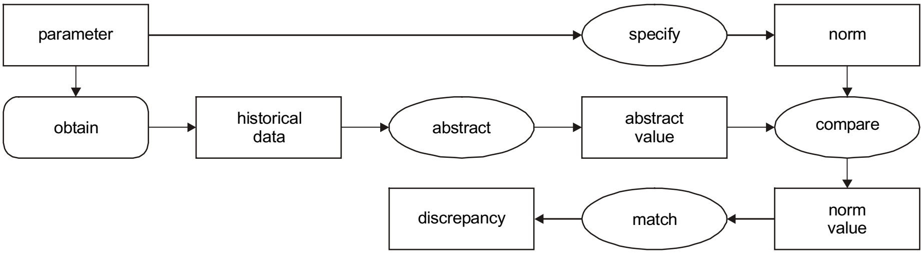
References

1. Alonso, C., Pulido, B., Acosta, G.: On Line Industrial Diagnosis: an attempt to apply Artificial Intelligence techniques to process control. In: 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-98. LNAI. Volume 1415., Springer-Verlag (1998) 804–813
2. Alonso, C., Acosta, G., Mira, J., Prada, C.: Knowledge based process control supervision and diagnosis: the AEROLID approach. Expert Systems with Applications 14 (1998) 371–383
3. Alonso, C., Pulido, B., Acosta, G., Llamas, C.: On-line Industrial supervision and diagnosis, knowledge level description and experimental results. Expert Systems with Applications 20 (2001) 117–132

4. Acosta, G., Alonso, C., Pulido, B.: Basic Tasks for Knowledge Based Supervision in Process Control. Engineering Applications of Artificial Intelligence 14 (2002) 441–455
5. Henao, M.: CommonKADS-RT: Una Metodología para el Desarrollo de Sistemas Basados en el Conocimiento de Tiempo Real. PhD Thesis, Univ. Pol. de Valencia (Spain) (2001)
6. Henao, M., Soler, J., Botti, V.: Developing a Mobile Robot Control Application with CommonKADS-RT. In: Engineering of Intelligent Systems. IEA/AIE 2001. LNAI vol. 2070., Springer (2001) 651–660
7. Selic, B., Bullekson, G., Ward, P.T.: Real Time Object-Oriented Modeling. Software Engineering Practice Series. John Wiley & Sons, New York (1994)
8. Selic, B., Rumbaugh, J.: Using UML for Modeling Complex Real-Time Systems. Whitepaper, ObjecTime Limited (1998)
9. Douglass, B.P.: Real-Time Uml: developing efficient objects for embedded systems. 3 edn. Addison Wesley Logman (1998)
10. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., Wielinga, B.: Knowledge Engineering and Management, The CommonKADS Methodology. The MIT Press (1999)
11. OMG: UML Profile for Schedulability, Performance and Time Specification. OMG Specification ptc/02-03-02 (2002)
12. Schreiber, G., Wielinga, B., Breuker, J.: KADS. A Principled Approach to Knowledge-Based System Development. Academic Press (1993)
13. Breuker, J., Van de Velde, W., eds.: CommonKADS Library for Expertise Modelling. Reusable problem solving components. Vol. 21 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam (1994)
14. Dressler, O., Struss, P.: The consistency based approach to automated diagnosis of devices. In: Principles of knowledge representation. CSLI publications, Stanford (1996) 269–314
15. Maestro, J.A., Llamas, C., Alonso, C.: Anotación de aspectos temporales en la especificación de la tarea en CommonKADS: Aplicación a la monitorización. In: IX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA-TTIA'01), Gijón, Spain (2001) 449–458
16. Burns, A., Wellings, A.: Real-Time Systems and Their Programming Languages. 3 edn. Addison-Wesley (2001)
17. Galton, A.: Time and Change for AI. In: Epistemic and Temporal Reasoning. Volume 4 of Handbook of Logic in Artificial Intelligence and Logic Programming. Oxford Science Publications (1995) 175–240
18. Alonso, C.J., Llamas, C., Maestro, J.A., Pulido, B.: Diagnosis of Dynamic Systems: A Knowledge Model that Allows Tracking the System during the Diagnosis Process. In: Developments in Applied Artificial Intelligence. IEA/AIE 2003. LNIA vol. 2718., Springer (2003) 208–218

Appendix

In this appendix, the analysis for the *intensive monitoring* task is described. It presents a task with similar time constraints specification as the *normal monitoring*.



```
TASK intensive-monitoring ;
  GOAL : "Analyze an ongoing process to find an abnormal
    behaviour" ;
  TYPE: periodic ;
  RELATIVE-TIME : Yes ;
  PERIOD : period ;
  ROLES :
    INPUT :
      parameter : "A signal which behavior is been
        analyzed" ;
      period : "Period which the task is realized" ;
    OUTPUT :
      discrepancy : "Any classification of abnormal
        behavior of the system being monitored" ;
  SPECIFICATION : "Watch the parameter evolution to
    confirm whether it behaves not according to system
    expectations" ;
END TASK

TASK-METHOD temporal-abstraction-monitoring ;
  REALIZES : intensive-monitoring ;
  DECOMPOSITION :
    INFERENCES : abstract, compare, match, specify ;
    TRANSFER-FUNCTIONS : obtain ;
  ROLES :
    INTERMEDIATE :
      norm : "Expected normal value for the parameter" ;
      abstract-value : "Some kind of abstracted result
        from historical data" ;
      historical-data : "Collection of past values for the
        parameter" ;
      norm-value : "Truth value that indicate whether a
        norm is fulfilled" ;
      norm-values : "Set of norm values" ;
    CONTROL STRUCTURE :
      obtain (parameter ! historical-data) ;
      WHILE HAS-SOLUTION
           abstract (historical-data → abstract-value) DO
        specify (parameter + abstract-value → norm) ;
        compare (abstract-value + norm → norm-value) ;
        norm-values = norm-values ADD norm-value ;
      END WHILE
      match (norm-values → discrepancy ) ;
END TASK-METHOD
```

Fig. 9. Inference structure and task description for the *intensive monitoring*.

On Description and Reasoning about Hybrid Systems

Katsunori Nakamura[1] and Akira Fusaoka[2]

[1] Information Media Center, Heian Jogakuin (St.Agnes') University
250 Miyake-cho Moriyama-city, SIGA, JAPAN 524-8511
[2] Department of Computer Science, Ritsumeikan University
Nojihigashi, Kusatsu-city, SIGA, JAPAN 525-8577

Abstract. In this paper, we introduce a nonstandard model of the situation calculus to deal with the hybrid system. The nonstandard situation calculus is build from the standard one via the ultra-product formation and it allows discrete but uncountable (hyper-finite) state transition, so that we can describe and reason about the interaction of the continuous and discrete dynamics. In this enlarged perspective of the nonstandard situation calculus, we discuss about the inherent problems to the hybrid dynamics such as ZENO problem.

1 Introduction

A hybrid system is a dynamical system in which continuous and discrete dynamics are interacting each other. For example, a real-time system is usually hybrid because it often contains the digital controller for the continuous environments. There are many difficult problems related to the hybrid system. For example, the verification of the correctness is necessary for the control program embedded in the automotive engine, but it is impossible to deal with the combustion dynamics in cylinder directly by the classical program verification. A planning is also a difficult task for a reactive agent situated in the physical world if any discrete sampling model for the continuous environment is insufficient. Therefore, a unified framework for both the discrete and the continuous dynamics are required for the formal treatment of hybrid systems.

There have been many works from the various fields for this problem. In the fields of artificial intelligence, a lot of efforts have been concentrated on the description of the continuous change in the discrete action-calculi [3][5][7][9]. In the situation calculus, for example, the situation with the duration is introduced to deal with the physical phenomena in which a Boolean fluent keeps the value but a continuous quantity may change. The behavior of continuous quantity is called the process and described as one of the fluents by using another description methods such as the differential equation or the continuous function of time. Therefore, the amalgamation of two description methods is used here: namely, the discrete state-transition by action and the differential equations to represent the continuous dynamics.

In the fields of the control theory and the automata theory, a hybrid automaton is proposed [1][4]. A hybrid automaton is the finite automaton that allows the continuous change of values with in a state. The continuous change itself is governed

[1] Currently, Ritsumeikan University

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 274-283, 2004.

by the set of differential equations labeled at each state and the state transitions are triggered whenever one of conditions labeled at the state is violated. Therefore, the hybrid automaton is an amalgamation of two description methods again: the finite automata for the discrete dynamics, and the differential equation for the continuous dynamics.

Although these formulations allow the treatment of the continuous change in the discrete paradigm, they are essentially weak because it is incomplete in the meaning that the limiting state cannot be defined for the infinite sequence of discrete states. For example, it is impossible to deal with so-called the clustered variation or Zeno problem[2], which is a typical interaction of the discrete and the continuous dynamics. The Zeno is a phenomenon in which the infinite iteration of a discrete value change occurs in the finite time, that is familiar not only in the industrial hybrid system but also in our daily life. For example, a bouncing ball shows the Zeno orbit in which a ball becomes to be rest in the finite time after the infinite iteration of bouncing (discontinuous change of velocity). It is often pointed out that the Zeno is a significant but hard problem not only from the standpoint of the knowledge representation [2] but also from the practical treatment of the hybrid system, so that the non-Zeno condition is usually prerequisite assumption at almost studies of hybrid system [4][10].

In this paper, we propose to use a nonstandard model of the situation calculus to deal with the hybrid system, in which the dynamics is described on the ontology of the hyperreals $^*R$ rather than R. In comparison with other methods such as hybrid automata or other action calculi, this method has the following advantages:

(1) Since the continuous change is defined by the sequence of the actions with the infinitesimal effect in the very small duration, we can deal with both the continuous and discrete dynamics uniformly in the discrete but hyper-finite state transition paradigm.

(2) The completeness in the space of discrete and continuous dynamics is naturally introduced so that it allows an asymptotic behavior toward limits such as Zeno.

(3) Since the all theorems of the standard situation calculus hold even in its nonstandard model (the transfer principle [6]), we can use the various properties of the discrete theory including the induction axiom even in its nonstandard extension [6].

2 Nonstandard Situation Calculus NSC

The situation calculus is a first-order logic with two sorts: situation sort s and action sort a. It contains a domain independent function $do(a,s)$, which gives the resulting situation when the action a is performed in the situation s. We use a real-valued fluent. The value of a fluent f at a situation s is denoted by $f(s)$, namely every fluent is treated as a function of the situation in this paper. We characterize a situation in the standard situation calculus **SSC** by a set of the independent fluent. Namely, a set of situation is :

$Sit = \{< f_1, f_2, \cdots f_m >| f_1 \in R, \cdots f_m \in R\}$. We use the special fluent "time *T*". We denote the set of action by Act. Every action has a duration τ where $\tau(a,s) = T(do(a,s)) - T(s)$.

A situation in **NSC** is constructed from the those of **SSC** via the ultra product formation [8].

Definition 1. Ultra filter: Let $\mathcal{F}$ be a family of the subsets of *N* which satisfy the following conditions, where *N* is the set of all natural numbers.

(1) $N \in \mathcal{F}$, $\varnothing \notin \mathcal{F}$

(2) if $A \in \mathcal{F}$ and $A \subset B$ then $B \in \mathcal{F}$

(3) if $A \in \mathcal{F}$ and $B \in \mathcal{F}$ then $A \cap B \in \mathcal{F}$

(4) for any $A \subset N$, $A \in \mathcal{F}$ or $N - A \in \mathcal{F}$

(5) if $A \subset N$ is finite then $N - A \in \mathcal{F}$

$\mathcal{F}$ is called an ultra filter.

Definition 2. Hyper-real: We fix an ultra filter $\mathcal{F}$. Let *W* denote a set of sequences of real numbers $(a_1, a_2, \cdots)$. The hyper-real number ${}^*R$ is defined by introducing the following equivalence relation into *W*

$$(a_1, a_2, \cdots) \sim (b_1, b_2, \cdots) \Leftrightarrow \{k \mid a_k = b_k\} \in \mathcal{F}$$

Namely, ${}^*R = W / \sim$. We denote the equivalence class of $(a_1, a_2, \cdots)$ by $[(a_1, a_2, \cdots)]$. We define a relation $a \approx b$ if the distance from *a* to *b* is infinitesimal. We distinguish the nonstandard variables and function symbols from the standard one by attaching * to them, although it is omitted in the clear cases. The continuity and differentiation of the *R*-valued function is defined in the ${}^*R$ in the following way.

Definition 3. Continuity: A standard function $f(x)$ is continuous at a standard number *x* if and only if for all $y \in {}^*R, f(y) \approx f(x)$ if $y \approx x$

Definition 4. Differentiation: A standard function $f(x)$ is differentiable at a standard number *x* if and only if there exists some $d \in R$ such that for every nonzero infinitesimal ε, $\dfrac{f(x+\varepsilon) - f(x)}{\varepsilon} \approx d$

Definition 5. Nonstandard situation: We fix an ultra filter $\mathcal{F}$. The situation in **NSC** is defined by

$$ {}^*Sit = \{< [(f_1^1, f_1^2, \cdots)], [(f_2^1, f_2^2, \cdots)], \cdots, [(f_m^1, f_m^2, \cdots)] > \\ \mid \{n \mid < f_1^n, f_2^n, \cdots f_m^n > \in Sit\} \in \mathcal{F}\} $$

From this definition, we can always find the limiting situation $[(s_1, s_2 \cdots)]$ of any sequence of standard situations $s_1, s_2 \cdots$ in ${}^*Sit$. Namely,

$$[(s_1, s_2 \cdots)] =< [(f_1^1, f_1^2, \cdots)], [(f_2^1, f_2^2, \cdots)], \cdots, [(f_m^1, f_m^2, \cdots)] >$$

where $s_n =< f_1^n, f_2^n, \cdots f_m^n >$ for each *n*.

Definition 6. Action and do function: The set of action Act is also enlarged to the set ${}^*Act$ by ${}^*Act = \{[(a_1, a_2, \cdots)] \mid a_i \in Act$ for each $i\}$

The *do* function is transferred to

${}^*do([(a_1, a_2, \cdots)], [(s_1, s_2, \cdots)]) = [(do(a_1, s_1), do(a_2, s_2), \cdots)]$

3 Description of the Dynamics

3.1 Infinite Iteration of Action

A dynamical system can be characterized generally as an infinite iteration of an action *a*, namely $(s, do(a, s), do(a, do(a, s)), \cdots)$. We denote the situation after repeating *a n* times from *s* by *iterate(a,n,s)*. In the following theorem, we prove that there always exists the nonstandard situation which is the result of infinite iteration of action.

Theorem 1. Infinite iteration of action

For any standard action $a \in Act$ and any situation $s \in Sit$,

$$iterate(a, \omega, s) \in {}^*Sit \quad \text{for } \omega = [(1,2,\cdots)].$$

Proof. Clearly, $iterate(a, \omega, s) = [(iterate(a,1,s), iterate(a,2,s), \cdots)]$.

by the definition. From the Definition3, $iterate(a, \omega, s) \in {}^*Sit$.

Definition 7. Fixed point: An action *a* has a fixed point if and only if

$iterate(a, \omega, s) \approx iterate(a, \omega + 1, s)$

A fixed point *s* is **attractive** if and only if there exists $Z \subseteq {}^*Sit$ such that $s \in Z$

$$\forall n \in N \forall s' \in Z[s' \neq s \wedge iterate(a, n, s') \in Z \wedge iterate(a, \omega, s') \approx s]$$

The fixed point *s* is **repelling** if and only if there exists $Z \subseteq {}^*Sit$ such that $s \in Z$

$$\exists n \in N \forall s' \in Z[s' \neq s \wedge iterate(a, n, s') \notin Z]$$

A fixed point $s = iterate(a, n, s_0)$ is **Zeno** related to *a*, *f* if and only if there exists an infinite subsequence $(i, j, \cdots, k, \cdots) \subseteq (1,2,\cdots)$ such that

$$\forall ij[f(s_i) \not\approx f(do(a, s_j)) \wedge time(s) \text{ is finite}]$$

3.2 Hyper-finite Recurrence Equation

In this paper, we treat the continuous dynamics in terms of hyperfinite recurrence equations. Therefore, we assume that an action with continuous motion can be divisible infinitely. This infinitesimal segment of action is called an infinitesimal action. The essential point of our method is that any standard action is equivalent to an infinite iteration of the infinitesimal action.

Definition 8. An Infinitesimal action: A nonstandard action α is called infinitesimal at the situation *s* if and only if its duration is infinitesimal, that is

$time(do(\alpha, s)) \approx time(s)$.

Theorem 2. Infinite division of action

If the standard action $a \in Act$ is n-divisible for any $n \in N$ then $div(a,\omega,k,s) \in {}^{*}Sit$ for each $k = 1,2,\cdots\omega$.

Proof. Let $\xi(n,k) = div(a,n,k,s)$ for finite n, k. We define nonstandard situations $\xi(n,k) = [(\xi(1,k_1),\xi(2,k_2),\cdots\xi(n,k_n),\cdots)]$, where $k_\omega = k$ and if $k_m < \dfrac{m}{2}$ then $k_{m-1} = k_m$ else $k_{m-1} = k_m - 1$.

By the definition 7, $\xi(\omega,k) \in {}^{*}Sit$. And also,

$$time(\xi(\omega,k)) = [(time(\xi(1,k_1)),time(\xi(2,k_2)),\cdots)] = time(s) + \frac{[(k_1,k_2,\cdots)]}{\omega}\tau$$

Therefore, we can conclude $\xi(\omega,k) = div(a,\omega,k,s)$.

We can define an infinitesimal action ∂a such that $do(\partial a, div(a,\omega,k,s)) = div(a,\omega,k+1,s)$ for each k. In the following, we denote the duration of ∂a by ν, that is $\nu = \dfrac{\tau}{\omega}$ for the duration τ of action a.

Clearly, $do(a,s) = iterate(\partial a,\omega,s)$. Namely, we can regard any standard action as an infinite iteration of the infinitesimal action. ∂a determines an orbit: $(u_1,u_2,,u_\omega)$ from s to $do(a,s)$ where $u_{i+1} = do(\partial a,u_i)$ for $0 < i < \omega$.

The differential equation can be easily transformed to the hyper-finite recurrent equation via ∂a. We explain how to deal with the differential equation in the next section.

4 The Description of Hybrid System

4.1 Alternative Water Tank

Consider a coupling of two water tanks [4]. Let x,y denote the level of water in Tank A and Tank B. We assume that the tap in the bottom of each tank discharges the water at a rate proportional to the level of each tank. Also the constant flow denoted by p of the water is poured exclusively to either Tank A (we call the state A) or Tank B (the state B) at each time (Figure 1,2). We use the control strategy.

if $st = A \wedge x \geq h \wedge y < h$ **then** switch to B

if $st = B \wedge y \geq h \wedge x < h$ **then** switch to A

The levels of water x,y are described below: (where st is the state of Tank)

if $st = A$ **then** $\dfrac{dx}{dt} + kx - p = 0 \wedge \dfrac{dy}{dt} + ky = 0$

if $st = B$ **then** $\dfrac{dx}{dt} + kx = 0 \wedge \dfrac{dy}{dt} + ky - p = 0$

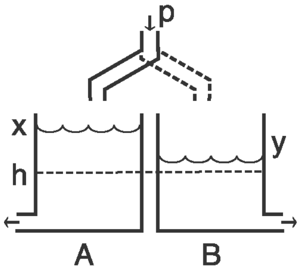


Fig. 1. Alternative water tank

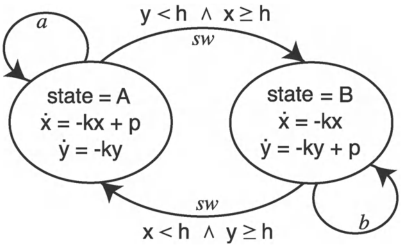


Fig. 2. State Transition

4.2 The System Description in NSC

We can describe this system in **NSC** as below.

[situation] <*st,x,y,t*> where

st: state {*A*,*B*}, *x*,*y*: level of each tank, *t*: time

[action] {*sw, a,b*} where

sw: state change, *a*,*b*: pouring the water in Tank A, B, respectively.

[Constant] *k,p,h,c*: standard numbers

ν : the minimal sampling time of the system ($\nu \approx 0$, but $\nu \neq 0$).

[Initial status] $x(s_0) = y(s_0) = c \wedge st(s_0) = A \wedge c > h$

[Effect axiom]

sw: $\forall s[st(s) = A \supset st(do(sw,s)) = B], \forall s[st(s) = B \supset st(do(sw,s)) = A],$
$\forall s[x(do(sw,s)) = x(s)], \forall s[y(do(sw,s)) = y(s)]$

a: $\forall s[x(do(\partial a,s)) = (1-k\nu)x(s) + p\nu, y(do(\partial a,s)) = (1-k\nu)y(s),$
$st(do(\partial a,s)) = st(s), do(a,s) = iterate(\partial a,n,s)]$

b: $\forall s[x(do(\partial b,s)) = (1-k\nu)x(s), y(do(\partial b,s)) = (1-k\nu)y(s) + p\nu,$
$st(do(\partial b,s)) = st(s), do(b,s) = iterate(\partial b,n,s)]$

[Control scenario]

$$\forall s_i \exists s_{i+1}[st(s_i) = A \wedge x(s_i) \geq h \wedge y(s_i) < h \supset s_{i+1} = do(sw,s_i),$$
$$st(s_i) = A \wedge (x(s_i) < h \vee y(s_i) \geq h) \supset s_{i+1} = do(a,s),$$
$$st(s_i) = B \wedge x(s_i) < h \wedge y(s_i) \geq h \supset s_{i+1} = do(sw,s_i),$$
$$st(s_i) = B \wedge (x(s_i) \geq h \vee y(s_i) < h) \supset s_{i+1} = do(b,s)]$$

4.3 Reasoning about System Behavior

By using this example, we present the inferential methods of temporal prediction for both continuous and discrete dynamics. The system contains Zeno. We prove its existence and localize the Zeno point, and discuss about how we can escape from it.

[the continuous dynamics]

Consider the differential equation of the action a:

$$\frac{dx}{dt}+kx=p, x(s_0)=x_0, time(s_0)=0$$

Let assume that we want to predict the situation s' after t seconds. We deal with this differential equation by using the piecewise constant model. The dynamics for the level of water are described by an action ∂a with duration ν.

For every situation u_i

$$x(do(\partial a,u_i))=(1-k\nu)x(u_i)+p\nu, time(do(\partial a,u_i))=time(u_i)+\nu$$

Namely, the do function defines the situations $u_i=iterate(\partial a,i,s_0)$ for $i=1,2,\cdots n$ where $n=\frac{t}{\nu}$. The desired situation s' is given by u_n. By Reiter's Induction rule [6], We have

$$x(u_n)=x(iterate(\partial a,n,s_0))=x_0\left(1-k\frac{t}{n}\right)^n$$

We must find out the standard value near $x({}^*u_n)$. We use a knowledge related to infinitesimal arithmetic: $\left(1-\frac{1}{n}\right)^n\approx e^{-1}$ if n is infinite. From this equation, we have

$$\left(1-k\frac{t}{n}\right)^n\approx e^{-kt}$$

Finally, we have the desired situation $s'=<x_0e^{-kt},t>$

Note that we use only common reasoning rules in the situation calculus and simple arithmetic for real and hyper-real numbers in the above argument. Namely, this suggests that the situation calculus is sufficient for the reasoning about hybrid dynamics if we deal with it in the nonstandard model.

[The existence of Zeno point]

The level of water x,y at the end of each state are described below.

$$x(s_1)=\frac{p}{k}+\frac{h\left(c-\frac{p}{k}\right)}{c}, y(s_i)=h \quad \text{where} \quad s_1=do(a,s_0)$$

Let $s_{2n}=iterate(a;b,n,s_0)$ and $s_{2n+1}=do(a,s_{2n})$. Then,

$$x(s_{2n+1})=\frac{p}{k}+\frac{h\left(h-\frac{p}{k}\right)}{y(s_{2n})}, y(s_{2n+1})=h, \tau(s_{2n+1})=\frac{1}{k}\log\left(\frac{y(s_{2n})}{h}\right)$$

$$x(s_{2n})=h, y(s_{2n})=\frac{p}{k}+\frac{h\left(h-\frac{p}{k}\right)}{x(s_{2n-1})},=h, \tau(s_{2n})=\frac{1}{k}\log\left(\frac{x(s_{2n-1})}{h}\right)$$

$$time(s_n)=\sum_{i=1}^{n}\tau(s_i)$$

We can prove the relations $h < x(s_{2n+1}) < x(s_{2n-1}), h < y(s_{2n}) < y(s_{2n-2})$ from the mathematical induction. Namely, the sequences of *x*,*y* are monotonously decreasing but bounded so that *x*,*y* have a limit point by the Weierstress's theorem. Namely, we have a standard $\hat{x}, \hat{y}, \hat{t}$ such that $x(s_{2n+1}) \approx x(s_{2n-1})$.

From $x(s_{2n+1}) \approx x(s_{2n-1}) \approx \hat{x}$ and a recurrence equation,

$$x(s_{2n+1}) = \frac{p}{k} + \frac{h\left(h - \frac{p}{k}\right)}{\frac{p}{k} + \frac{h\left(h - \frac{p}{k}\right)}{x(s_{2n-1})}}$$

we have $\hat{x} = h$. Similarly, we can get $\hat{y} = h$. On the other hand, we have

$$(x(s_{n+1}) + y(s_{n+1})) = (1 - k\nu)(x(s_n) + y(s_n)) + p\nu$$

from the system description. By solving this recurrence equation, we have $\hat{t} = \frac{1}{k}\log\left(\frac{2ck - p}{2hk - p}\right)$. $\hat{t}$ is finite so that we can prove that the situation

$\hat{s} =< h, h, \frac{1}{k}\log\left(\frac{2ck - p}{2hk - p}\right) >$ is Zeno point.

[Escape from Zeno]

This Zeno point is repelling for $t > \hat{t}$ so that the behavior of the system at $t > \hat{t}$ depends on the situation immediately after the Zeno point. Since the duration of any action should not be smaller than the minimal sampling time ν, the action *a* continues during ν when τ_n becomes smaller than ν at the situation:

$st(s_{2n+1}) = A \wedge h(1 - k\nu) \geq y(s_{2n})$. Then the next value of *y* becomes lower than *h* and the next system time is over $\hat{t}$ (See Figure 3). So we can jump out from the fixed point

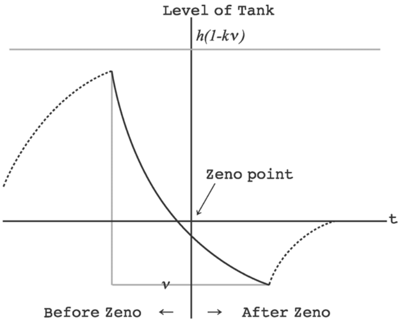


Fig. 3. Monad around Zeno point

The Figure 4 shows a numerical simulation result of this system. We use some very small numbers for v, and the other parameters are

$x(s_0) = 2.0 \wedge y(s_0) = 1.5 \wedge k = 0.5 \wedge h = 1.0 \wedge p = 0.9$.

The excursion of the level of tank is contained in the envelope

$$w = \left(2c - \frac{p}{k}\right)e^{-kt} + \frac{p}{k}$$

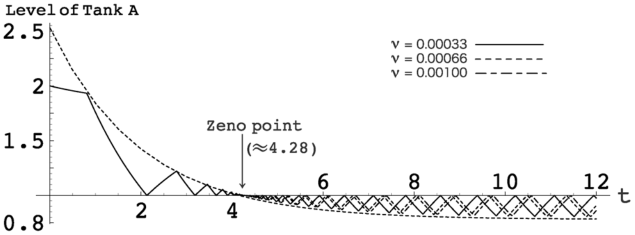


Fig. 4. The orbit of the water level of Tank A

5 Concluding Remarks

We need the following inferential device in order to deal with the hybrid dynamics actually:

(1) a reasoning system for the situation calculus

(2) an extended arithmetic for $^*R$

(3) a set of transfer rules between $^*R$ and R such as $\left(1 - \frac{1}{n}\right)^n \approx e^{-1}$ if n is infinite.

We also need the data structure and evaluation mechanism for the class of non-standard number.

The implementation of the reasoning system and the symbolic simulator for hybrid system are currently under development.

References

1. Alur,R., Courcoubetis,C. Henzinger,T.A, Ho, P.: Hybrid Automata: An algorithmic Approach to the Specification and Verification of Hybrid Systems. Hybrid Systems (Grossman, R.L., et al. eds) LNCS 736 (1993) 209-229
2. Davis, Y.: Infinite Loops in Finite Time: Some Observations, Proceedings KR-92, (1992) 47-58
3. Herrmann,C.S. and Thielscher,M.: Reasoning about Continuous Processes, Proceedings AAAI-96 (1996) 639-644

4. Johansson, K.H.,Egerstedt,M.,Lygeros,J. and Sasty,S. : On the Regularization of Zeno Hybrid Automata, System & Control Letters 38. (1994) 141-150
5. Miller, R. and Shanahan, M: Reasoning about Discontinuities in the Event Calculus. Proceedings. KR 96 (1996) 63-74
6. Reiter,R.: Proving Properties of States In The Situation Calculus. Artificial Intelligence 64(1993) 337-351
7. Reiter, R. : Knowledge in Action, Cambridge: The MIT press (2001)
8. Robinson, A. : Non-standard analysis. Amsterdam: North Holland (1974)
9. Thielscher, M.: The logic of Dynamic System. Proceedings of IJCAI (1995) 1956-1962.
10. Zhang, J., et al. : Dynamical System Revisited: Hybrid Systems with Zeno Execution, Proceedings HSCC2000. LNCS 1790 (2000) 451-464

A Hybrid Approach to Automatic Word-Spacing in Korean

Mi-young Kang, Sung-woo Choi, and Hyuk-chul Kwon

Korean Language Processing Lab, School of Electrical & Computer Engineering,
Pusan National University, San 30, Jangjeon-dong
609-735 Busan, Korea
{kmyoung, saebi, hckwon}@pusan.ac.kr
http://klpl.re.pusan.ac.kr/

Abstract. This paper proposes a hybrid automatic word-spacing system for the Korean language, combining stochastic- and knowledge-based approaches. Our system defines the optimal splitting points of an input sentence using two simple parameters: (a) relative word frequency and (b) Syllable n-gram statistics, extracted from large processed corpora that contain 33,643,884 word-tokens. Whereas this method efficiently resolves problems due to eventual data noise using processed training data, and data sparseness using Syllabic n-gram statistics and large corpora, there still remains the problem of processing unseen words, which can hardly be overcome even with a huge corpus. Therefore, this study compensates for the stochastic-based approach, (a) dynamically expanding candidate words with longest-radix selection among possible morphemes and (b) adopting inequivalent treatment between major lexical categories and minor lexical categories. The current combined model remedies drawbacks of the stochastic-based word-spacing algorithm and shows encouraging results: it obtained 97.51% precision in word-unit correction from the external test data.

1 Introduction

Word-spacing[1] errors in Korean need to be processed regarding Korean language typology and normative grammar. The present study is an attempt to implement a word-spacing method using simple statistical and linguistic information to the extent that its application does not burden the system. Spacing errors produce confusion in interpretation of parts of speech,[2] and thus linguistic errors and ambiguities are result.

[1] A Korean word can be composed of one morpheme or several concatenated morphemes of different linguistic features which are equivalent to a phrase in English. This spacing unit is referred to as a □word, □eo-jeol□or □morpheme cluster□in Korean linguistic literature. In this paper, we adopt □word□in order to refer to □an alphanumeric cluster of morphemes, located between two blanks in Korean□

[2] The word-spacing rules are given in the normative Korean grammar (Revised Korean Spelling System and Korean Standard, 1989) as follows.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 284-294, 2004.

Processing word-spacing, therefore, is crucial to Korean text processing. For example, in the information retrieval system, a morphological analysis of query words or phrases fails if there is a spacing error. Besides, the correct conversion of Korean text to phoneme for developing Text-to-Speech Synthesis (TTS) is impossible when there are spacing errors in the text. According to the Korean spacing constraints, a postposition and a prefix should be attached to the noun without a space, whereas an adverb appears with a space on its left and right sides. [3], [4]

gug-boda # bab-i # nas-da soup-POST /rice-NOM /to be better-END *"Rice is better than soup."* (1)

boda # nop-eun # isang more(ADV) /to be high-REL /ideal-NOM *"The lofty ideals"* (2)

The homonym, *boda* can be a postposition in (1) as well as an adverb in (2). Its grammatical categories are defined by spacing on the morphological level, without analyzing the syntactic features of its contexts. The following example shows the violation of this constraint.

*gug # boda # bab # nas-da soup /more(ADV) /rice-NOM /to be better-END (3)

In (3), because the morpheme *boda* is located between two spaces, it should only be interpreted as an adverb. An adverb cannot appear between nouns. As such, the sentence (3) becomes ungrammatical. A large number of Korean word-spacing rules are optional or difficult to memorize. For example, the spacing between compound nouns or between verb and auxiliary verb sequences is optional. Korean writers violate them intentionally or unintentionally. It makes Korean language processing more complicated.

In addition, according to a change of spacing, syntactic and semantic interpretations, and pronunciation, differ as well. This emphasizes the necessity of text preprocessing for TTS.

gug-bab soup-rice *'boiled rice soup'* /kukpap/ → [kukp□ap] (4)

gug # bab soup /rice *'soup (and) rice'* /kuk/ # /pap/ → [kuk] # [pap] (5)

Tensification is one of the phonological phenomena of the Korean language. It refers to the phenomenon in which lenis consonants such as /p/, /t/, /k/, /s/ and /c/ are changed to tense consonants [p□], [t□], [k□], [s□] and [c□] respectively, under certain

a. A dependent noun appears after a determiner with a space in a sentence.
b. An adverb appears in a sentence with a space.
c. A determiner is attached to the noun with a space on both sides.
d. A postposition and a prefix are attached to a noun without a space.
e. A suffix is attached to a verbal stem without a space.

[3] The symbols and abbreviations used for simplification in this paper are as follows. |: separation; #: spacing (word bound); -: morpheme boundary; *: unacceptable form; **[ ]**: phonetic representation; / /: phonological representation or morphological analysis; **( )**: facultative element; **ADV**: adverb; **END**: verbal ending; **NOM**: nominative; **POST**: postposition.

[4] We adopt the Korean standard for Romanizing Korean script and International Phonetic Alphabet for phonetic representation.

phonological or morphological conditions. The phonological condition is known as the post-obstruent tensing effect in inner-word structures. The consonant /p/ is changed to [p□] by post-obstruent tensing in (4), while it has no effect in (5) because, due to spacing, the /p/ in question is not a part of an inner-word structure.

This paper proposes a hybrid model combining a knowledge-based approach and the stochastic approach in order to construct an automatic word-spacing system based on stochastic information (i.e. the stochastic information of words and syllable n-grams) extracted from a large omnibus corpus. The system determines the most probable word-spacing point with the Viterbi algorithm, mainly using two parameters (i.e. word probability and the odds in favor of the inner-spacing of a given disyllable). This paper proposes, in order to overcome any problems due to data sparseness and training-data-dependency, a sequence of smoothing methods incorporating a knowledge-based approach which uses (a) the □longest match strategy□and a dynamic splitting-point selection method based on the viable prefix and (b) an unequal weight endowment strategy for (i) major lexical categories (content words) such as noun, verb, adjective and adverb, and (ii) minor lexical categories (grammatical morphemes) such as noun suffix, verbal pre-ending, verbal ending, and others. Our method of combining the knowledge-based approach with the stochastic approach proceeds so that the knowledge-based application does not burden the system.

This paper is composed of six sections. Following this introduction, section 2 presents previous studies concerning automatic word-spacing methods. Section 3 describes an automatic word-spacing algorithm using syllable n-grams and word stochastic information. Section 4 presents our automatic word-spacing algorithm combining a knowledge-based approach with the stochastic approach. In section 5, we discuss the results of experiments. Finally, concluding comments are given together with suggestions for future work.

2 Related Studies

Previous work on word-spacing can be classified into (a) rule- and knowledge-based approach and (b) the stochastic approach. Among rule- and knowledge-based approaches, Sim CH.M. et al. developed heuristics, a □longest match strategy□, and a priority application based on the morphological analysis. [9] And Kim S.N. et al. applied syllabic lexical-combination rules to improve system performance by 95.75 %. [6] Kang S.S. considered the syllabic properties of postpositions or endings, applied a morphological analysis within a word block in order to find a word boundary and obtained a 93.2% word-unit recall and a 97.3% syllable-unit recall. [5] Kang M.Y. and Kwon H.CH. improved spacing correction for erroneous words that should be split into more than two words by investigating rules and clues, and obtained a precision of 98.01% in spacing-error correction. [2] These previous rule- and knowledge-based approaches have shown a high accuracy in processing word-spacing using linguistic features. Nevertheless a disadvantage is the amount of language that has to be treated without the possibility of expanding this knowledge base and putting it to practical use. Furthermore, these approaches require time-consuming and labor-intensive work.

Contrary to rule-based methods, the stochastic method has advantages in set-up time and cost savings and capability of coping with unregistered words. Among stochastic approaches we can firstly enumerate that of Shim K.S.. This study used syllabic properties on the basis of the corpus-based statistical information. [10] Kang S.S. and Woo C.W. proposed space-insertion probability $P(x_i, x_{i+1})$ for adjacent disyllables x_ix_{i+1} in a given input sentence such as $x_1x_2 \ldots x_ix_{i+1} \ldots x_{n-1}x_n$. Space insertion is estimated by considering left-, right- and inner-space probabilities extracted from the raw corpus. The study showed a 97.7% syllable-unit accuracy, 90.5% for word-spacing problems at the end of a line, and 82.1% for word-spacing error detection. [4] Lee D.G. et al. treated word-spacing problems such as POS tagging, using a hidden Markov model (HMM), and found the most likely sequence of word-spacing tags $T = (t_1, t_2, \ldots, t_n)$ for a given sentence of syllables $S = (s_1, s_2, \ldots, s_n)$ with the equation: $\arg\max_{T} p(T|S)$. The model ^ *(T(2:2), S(1:2))*, which uses a syllable tri-gram, gave the best results: 98.33% correctly spaced syllables compared to the total number of syllables in the document and 93.06% correctly spaced words compared to the total number of words created by the system. [7] Nevertheless, the stochastic methods are revealed to involve a strong training-data-dependency and data sparseness. Data sparseness becomes more serious while processing agglutinative languages such as Korean. Because in the agglutinative languages, chains of particles or endings are attached to the ends of nominal or verbal roots and determine most of the grammatical relations rather than express the semantic content as separate words, there is considerable risk in using syllabic statistics for the right-hand boundary of a word. Each particle can appear alone or with other particles in succession. Thus, we cannot cover all the variation probabilities of a Korean word even with a huge corpus.

3 Syllable N-Grams, Word Statistics, and Word-Spacing Algorithm

An efficient automatic word-spacing system can be constructed with simple statistical information extracted from training data. Here we should consider the training-data size in order to face data sparseness, and we need also to consider the training-data quality in order to overcome data-dependency. Our stochastic system is constructed with two simple parameters on the basis of the □Word probability□and the □Inner-spacing probability of a given disyllable□extracted from large corpora without training data noise.

3.1 Training Data and Stochastic Information

A stochastic method shows a high data-dependency. It is very important to use training data with the least possible noise in order to extract words and Syllabic n-gram stochastic information. To satisfy this condition, we use different corpora which were edited for word-spacing by standard Korean grammar experts. The following chart shows the composition of the edited corpora containing 1,950,068 word-types and 33,643,884 word-tokens.

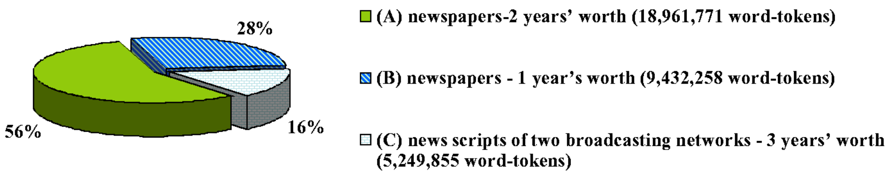


Fig. 1. Processed Training Data (33,643,884 Word-Tokens)

3.2 Word Probability

The probability that a given sequence could be a possible single word (i.e. word probability), $P(w_i)$, is measured simply by relative word frequency.

$$P(w_i) = \frac{freq\ (w_i)}{\sum_{i=1}^{n} freq\ (w_i)} \quad where,\ n = total\ number\ of\ word\text{-}types \qquad (6)$$

For the word sequence $w_i w_{i+1}$, in which spacing is optional (i.e., compound nouns and verb and auxiliary verb sequence), each probability of w_i, w_{i+1}, and the compound $w_i w_{i+1}$ are set as 0.5:

$$P(w_i) = P(w_{i+1}) = P(w_i w_{i+1}) = 0.5. \qquad (7)$$

3.3 Syllabic N-Gram Statistics

Syllabic n-gram frequencies are sorted according to their left-, inner- or right-spacing aspects. We do not use the probability of a right-space for a given syllable n-gram because Korean is an agglutinative language in which sequences of particles or endings are commonly attached to the ends of nominal and verbal roots, and thus there is risk in using syllable statistics for the right-hand boundary. This is the reason why Korean is especially vulnerable in relation to data sparseness.

Odds in Favor of the Inner-spacing of a Disyllable. An automatic word-spacing system based only on a word probability cannot resolve the data sparseness problem. Our system, thus, provides the inner-spacing probability of a disyllable. The inner-spacing probability of a given disyllable is estimated by the following equation which is constructed using the relative inner-spacing frequency of the disyllable:

$$P_{innerS}(x,y) = \frac{freq(x\_y)}{freq(xy) + freq(x\_y)} \quad where,\ `\_' = space. \qquad (8)$$

The odds favoring the inner-spacing of a given disyllable, $Odds_{innerS}(x,y)$, is estimated by the rate of the inner-spacing probability of the disyllable, $P_{innerS}(x,y)$, compared to the rate of no-inner-spacing probability, that is, 1-P_{innerS}(x,y):

$$\beta = Odds_{innerS}(x,y) = \frac{P_{innerS}(x,y)}{1 - P_{innerS}(x,y)}. \tag{9}$$

Syllabic N-Gram Statistics for Pre-chunking. Together with the inner-spacing probability of a disyllable in equation (8), the left-spacing probability of a syllable n-gram, $P_{leftS}(x)$ or $P_{leftS}(xy)$, can be used for the pre-chunking of given input data. This reduces the burden on the system. If the value of $P_{innerS}(x,y)$, and the $P_{leftS}(x)$ or $P_{leftS}(xy)$ value are 1, unigram *x* and disyllable *xy* always appear with a left space. The obligatory splitting conditions are:

$$P_{innerS}(x,y) = \frac{freq(x\_y)}{freq(xy) + freq(x\_y)} = P_{leftS}(x) = \frac{freq(\_x)}{freq(x)} = P_{leftS}(xy) = \frac{freq(\_xy)}{freq(xy)} = 1. \tag{10}$$

When this value is 0, the syllable n-gram is always without an inner space.

3.4 Stochastic Word-Spacing Algorithm

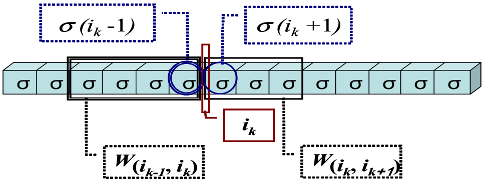
The optimal word-spacing points are obtained by finding the optimal *S*, which denotes a sequence of possible words or a possible sentence:

$$S = \{w_{(i_0,i_1)}, w_{(i_1,i_2)}, \cdots, w_{(i_{n-1},i_n)}\}. \tag{11}$$

i_k: k^{th} spacing location; i_0: initial of *S*;

$w_{(i_{n-1},i_n)}$: candidate word located between i_{n-1} and i_n.

Fig. 2 shows candidate words and word-boundary syllable. Spacing probabilities are estimated by the using maximum likelihood estimator, which entails two simple parameters: (a) the k^{th} word probability, $p(w_{(i_{k-1},i_k)})$, and (b) the odds favoring the inner-spacing of a given disyllable, $Odds_{innerS}(xy)$, at the candidate word boundary, i_k.



i_k-1: the location of the syllable immediately preceding i_k
i_k+1: the location of the syllable immediately following i_k
$\sigma(i_k -1)$: the final syllable of $W(i_{k-1}, i_k)$
$\sigma(i_k+1)$: the initial syllable of $W(i_k, i_{k+1})$

Fig. 2. Candidate Words and Word-Boundary Syllables

The optimal *S* is found by using the Viterbi algorithm. This algorithm computes a value for every possible word set, *S*, and chooses the one with the largest value:

$$\arg\max_{S} \sum_{k=1}^{n} \{\log p(w_{(i_{k-1},i_k)}) + m \log \beta\} \tag{12}$$

where, m = 2.42; $\beta = \frac{p_{innerS}(\sigma_{(i_k-1)}, \sigma_{(i_k+1)})}{1 - p_{innerS}(\sigma_{(i_k-1)}, \sigma_{(i_k+1)})}$; *If k = n,* $p_{innerS}(\sigma_{(i_k-1)}, \sigma_{(i_k+1)}) = o.5.$

Computing logarithms of equation (12) avoids underflow. In addition, multiplying the β value by the exponent of a power m, which is obtained by examining the inner test data, yields the best spacing accuracy. If the values of two *S*s are the same, then we choose one of them arbitrarily.

4 Stochastic- and Knowledge-Based Word-Spacing System Syllable

In this chapter we will show that an automatic word-spacing algorithm can be performed with the above stochastic-based algorithm, equation (12), using simple word frequency and syllable bi-gram statistics. And, as it is unlikely that this algorithm will resolve the word-spacing without the data sparseness problem, this study attempts to dynamically expand the candidate word list with a combined model.

4.1 Simple Stochastic Word-Spacing (SSWS) Model

Our SSWS model is applied in three steps. In the first step, the system defines obligatory splitting points according to the left-side spacing probability of syllable n-grams, P_{lefS}(x) and P_{lefS}(xy), and the inner-spacing probability of the disyllable, P_{innerS}(x,y). If these probabilities are 1, the system defines the splitting point and splits S further before proceeding to the next step.

In the second step, equation (12) is applied. Automatic spacing fails when the value of $Odds_{innerS}$(x,y) is 0. By examining the test data, we extend the range of the failure condition of the automatic spacing until a threshold at which it ceases to contribute to the system□s accuracy. If the value of $Odds_{innerS}$(x,y) is smaller than the threshold, the automatic spacing of a given random S including the syllable pair xy is passed over to the next automatic spacing step. In this step, a monosyllabic word requires special treatment.[5]

In the third step, all the S which failed to be spaced automatically in the previous steps, are processed. We obtained the threshold for the obligatory splitting point as 0.71 by examining the test data. When the value of $Odds_{innerS}$(x,y) is larger than the threshold value, the spacing point is given between syllables x and y. Equation (12) is applied again to the obtained blocks in this way.

4.2 Combined Word-Spacing (CWS) Model

The simple stochastic-based word-spacing method, above, resolves efficiently problems due to eventual data noise using processed training data, and resolves data

[5] Possible monosyllabic word probabilities can cause suffix- and postposition-spacing. Therefore when a monosyllabic word is included in the set S, our system estimates the failure value by calculating the number of a given syllable unigram appearing as a word compared to the total number of a given syllable unigram in the test data. If the value is smaller than 0.003, the automatic spacing of a given S including the monosyllabic word is passed over to the next step.

sparseness problems using syllabic n-gram statistics and large corpora. Nevertheless, there still remains the problem of processing unseen words which even a huge corpus can hardly overcome. This study compensates for the drawbacks of the stochastic-based approach by (a) dynamically expanding the number of candidate words by selecting according the longest-radix among possible morphemes and (b) adopting inequivalent treatment between major lexical categories and minor lexical categories.

Our basic system provides a list of possible words with their relative frequencies. A *dynamic expansion of word-list* (DEWL) uses a □longest match strategy□based on the viable prefix. This longest-radix search strategy suggests dynamically possible words and includes them among possible k^{th} candidate words by assigning a heuristic probability value (1.0/1billion) to them. The DEWL model efficiently compensates for the SSWS model which provides a simple data-sparseness-compensation-parameter (i.e. odds in favor of the inner-spacing of a disyllable), by using the inner-spacing probability of the disyllable which is located at the current candidate words boundary.

An inequivalent treatment between major lexical categories and minor lexical categories supplies an efficient strategy to compensate for data sparseness resulting from agglutinative morphology by providing productive inflectional and derivational suffixations. Inequivalent treatment includes unequal weight endowment strategies for major lexical categories and minor lexical categories. The system provides a *preferential non-spacing-value* (PNSV) for minor lexical categories which should be attached to a stem. If a given k^{th} candidate word is determined to be of a minor lexical category, the system assigns it double the heuristic probability-value of a normal longest-radix word (1.0/0.5billion).

Another inequivalent treatment is *dynamic selection strategy* (DSS) between major lexical category morphemes and longest-radix words (the minor lexical categories are excluded from the dynamic selection).

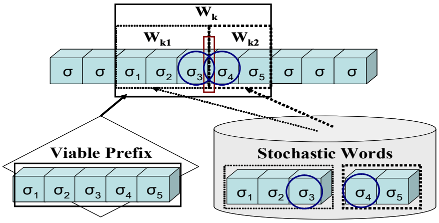


Fig. 3. Competition between Longest-radix and Stochastic-based Candidate Words

When a k^{th} candidate word (W_k) as shown in Fig. 3 is dynamically proposed by the □longest match strategy□and its inner chunks (stochastic-based words W_{k1} and W_{k2}, as shown in Fig. 3) are provided as constituting possible words by word probability, the system selects an optimal word by DSS. The system selects a candidate word by estimating the inner-spacing probability value of the disyllable located at the boundary of stochastic-based words (as shown, σ_3 and σ_4). If this value is under the threshold which is set as 0.3, the system selects the word from the stochastic candidate-words and, if the value is over the threshold of 0.7, the system selects the longest-radix word.

5 Experimentation

The test data was extracted from our training corpora (innerTD) and 21st Centry Sejong Project□s raw corpus[6] (externalTD). The inner test data (innerTD) was extracted according to the same distribution ratio as a given corpus in the whole training corpora (see Fig. 1).

Table 1. Test Data

Test Data		N° of Sentences	N° of Words	N° of Syllables
innerTD	A(56%) + B(28%) + C(16%)	2,000	17,396	65,483
externalTD		2,000	13,971	40,353

We obtained the input for the experiment by modifying the test data by removing spaces. The following results were obtained by using four kinds of evaluation measures. These include (a) syllable-unit precision, P_{syl}; (b) word-unit recall, R_w; (c) word-unit precision, P_w; and (d) setence-unit precision, P_S:

$$P_{syl} = \frac{Correctly spaced\ syllables}{Total\ spaced\ syllables} \times 100\ (\%) \quad P_w = \frac{Correctly\ spaced\ words}{Total\ spaced\ words} \times 100\ (\%) \tag{13}$$

$$R_w = \frac{Correctly spaced\ words}{Total words\ in\ test data} \times 100\ (\%) \quad P_S = \frac{Correctly spaced\ sentences}{Total\ spaced\ setences} \times 100\ (\%)$$

Table 2 shows the accuracy of the automatic spacing system using only the SSWS model.

Table 2. System Performance of SSWS (%)

Test Data	P_{syl}	P_w	R_w	P_S
innerTD	99.48	98.21	97.91	87.40
externalTD	97.83	90.84	94.10	71.10

Table 3 compares the system performance, increasing progressively according to combination range variation between the stochastic- and knowledge-based models.

[6] This balanced corpus (http://www.sejong.or.kr/english/index.html) includes various genres of Korean literature which have been used since the early 20th century. Many Korean language-processing studies use this corpus in order to measure their system□s performance, as it is designed to reflect the lexical changes which have occurred according to the changing social conditions of modern times and the way that the language is used by each age group, class, academic discipline, and interest, in society.

Table 3. System Performance of Combined Models (%)

CWS Models	Test data	P_{syl}	P_w	R_w	P_S
SSWS + DEWL	innerTD	99.50	98.36	97.93	88.20
	externalTD	90.30	97.33	97.77	90.05
SSWS + DEWL + PNSV	innerTD	99.50	98.39	97.94	88.40
	externalTD	99.31	97.49	97.76	90.40
SSWS + DEWL + PNSV + DSS	innerTD	99.50	98.39	97.93	88.20
	externalTD	99.31	97.51	97.77	90.45

Even though the SSWS model shows rather a high performance with the inner-test data (see Table 2), it does not perform as well with the external data as the combined model (see Table 3). This inequivalence results from the data sparseness. As shown in the Table 3, the average amelioration of word-unit-correction precision was progressive according to the degree of combining the knowledge- and stochastic-based models. The accuracy of the system improves the aiding of candidate words using the longest-radix search model. The accuracy improves further from guessing the candidate word by unequal weight endowment strategies for (i) major lexical categories and (ii) minor lexical categories: (a) the preferential non-spacing-value for the candidate ending and (b) the dynamic selection strategy between major lexical category morphemes and longest-radix words. The performance improves for the external data while not decreasing that for the inner data: with the combined model, the amelioration of word-unit-correction precision was about 6.67% for external test data. The system thus becomes robust against unseen words.

6 Conclusions and Further Studies

The objective of this study was to implement a robust automatic word-spacing system which is simple and robust against unseen words while processing word-spacing. This study uses a stochastic-based approach and compensates for it by dynamically expanding candidate words with longest-radix selection among possible morphemes and by adopting (a) a preferential non-spacing-value for the candidate ending and (b) a dynamic selection strategy between major lexical category morphemes and longest-radix words. Our combined model (SSWS + DEWL + PNSV + DSS) remedies the shortcomings of the simple stochastic-based word-spacing algorithm and shows positive results: the amelioration of word-unit-correction precision was about 6.67% with the combined model. An encouraging fact is that a similar performance was observed with the inner test data and the external test data: 98.39% and 97.51% precision in word-unit correction was observed, respectively. This result proves that the current system remains stable against data sparseness.

Though we could in large part resolve data sparseness, there still remain some problems. First, even though our current system dynamically provides candidate words, it hardly covers neologisms and proper nouns which appear in newspapers at a high frequency. Second, many words appear with the same probabilities and cause semantic and syntactic ambiguities. In order to resolve these linguistic ambiguities and cope with language productivity, our further work aims to develop a predictive algorithm for unseen words, and to refine the optimal combining algorithm composed of the statistical spacing method and the rule- and knowledge-based spacing method.

Acknowledgements. This work was supported by National Research Laboratory Program (Contract Number: M10203000028-02J0000-01510).

References

1. Chung Y.M., Lee J.Y.: Automatic Word-segmentation at Line-breaks for Korean Text Processing, Proceedings of 6th Conference of Korean Society for Information Management (1999) 21-24
2. Kang, M.Y., Kwon, H.CH.: Improving Word Spacing Correction Methods for Efficient Text Processing, Proceedings of the Korean Information Science Society, (B) Vol. 30. 1 (2003) 486-488
3. Kang, M.Y., Choi S.J., Yoon A.S., Kwon H.CH.: Stochastic Word-Spacing System with Dynamic Increase of Word List, to appear in *Proceeding of the First International Joint Conference on Natural Language Processing* (2004)
4. Kang, S.S.: Automatic Segmentation for Hangul Sentences, Proceeding of the 10th Conference on Hangul and Korean Information Processing (1998) 137-142
5. Kang, S.S., Woo C.W.: Automatic Segmentation of Words Using Syllable Bigram Statistics. Proceedings of 6th Natural Language Processing Pacific Rim Symposium (2001) 729-732
6. Kang, S.S.: Korean Morphological Analysis and Information Retrieval, Hongleunggwahag Publisher, Seoul (2002)
7. Kim, S.N., Nam, H.S., Kwon, H.CH.: Correction Methods of Spacing Words for Improving the Korean Spelling and Grammar Checkers, Proceedings of 5th Natural Language Processing Pacific Rim Symposium (1999) 415-419
8. Lee, D.G., Lee, S.Z., Lim, H.S., Rim, H.CH.: Two Statistical Models for Automatic Word Spacing of Korean Sentences, Journal of KISS(B): Software and Applications, Vol. 30. 4 (2003) 358-370
9. Manning, C.D., Scht□e, H.: Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge London (2001)
10. Sim, CH.M., Kwon, H.CH.: Implementation of a Korean Spelling Checker Based on Collocation of Words, Journal of KISS(B): Software and Applications, Vol. 23. 7 (1996) 776-785
11. Sim, K.S.: Automated Word-Segmentation for Korean Using Mutual Information of Syllables, Journal of KISS(B): Software and Applications, Vol. 23. 9 (1996) 991-1000

Natural Language Requirements Analysis and Class Model Generation Using UCDA

Dong Liu[1], Kalaivani Subramaniam[1], Armin Eberlein[2], and Behrouz H. Far[1]

[1] Department of Electrical and Computer Engineering, University of Calgary, 2500,
University Drive, N.W., Calgary, Alberta, Canada, T2N 1N4
{liud, subrama, far}@enel.ucalgary.ca
[2] Computer Engineering Department, American University of Sharjah, UAE
eberlein@enel.ucalgary.ca

Abstract. This paper presents a methodology to automate natural language requirements analysis and class model generation based on the Rational Unified Process (RUP). Use-case language schemas are proposed to reduce complexity and vagueness of natural language. Some rules are identified and used to automate class model generation from use-case specifications. A CASE tool named Use-Case driven Development Assistant (UCDA) is implemented to support the methodology. UCDA can assist the developer to generate use-case diagrams, use-case specifications, robustness diagrams, collaboration diagrams and class diagrams in IBM Rational Rose. It helps accelerate requirements analysis and class modeling, and reduce the time to market in software development.

1 Introduction

Object-Oriented Analysis and Design (OOAD) has become a very popular software development approach since the 1990's. Object elicitation and class modeling are among the central activities in OOAD. The objects are identified from the requirements, and the class model is generated based on them as well. Generally, there are two ways to specify the requirements: using formal languages or using natural languages (NL). The research community has focused on methods based on formal language requirements [1-3], while NL is widely used for requirements documentation in industry. It is hard to automate NL requirements analysis, because NL is inherently complex, vague and ambiguous [4].

Most commercial CASE (Computer Aided Software Engineering) tools do not supply the functionality of NL requirements analysis. However, there are several such tools that have been developed for research. CoGenTex Inc. developed a prototype tool named LIDA (Linguistic assistant for Domain Analysis), which provides linguistic assistance in model development [5]. The tool can process textual documents and help the user to generate a class model visualized in UML (Unified Modeling Language). NIBA (Natural Language Requirements Analysis in German) is an interdisciplinary project between computer scientists and computer linguists at the University of Klagenfurt, Austria [6]. The tool can parse requirements documents in German, interpret and transform output of the parser to conceptual pre-design

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 295-304, 2004.

schemas, validate the schemas and finally generate a conceptual model in UML. These approaches only generate the conceptual model, but the behavior of classes still need to be identified separately.

So far, it is impossible for machines to automatically perform the whole OOAD process, but it is possible to automate some micro-activities in it. We developed a method for use-case model generation, object identification and class modeling with respect to natural language requirements based on the Rational Unified Process (RUP). The methodology is introduced in Sect. 2. Sect. 3 presents the CASE tool that was developed based on the methodology. Sect. 4 concludes the paper.

2 Methodology

Based on the Rational Unified Process (RUP), the activities and corresponding artefacts during requirements, analysis and design are specified as follows:

- Identify actors and use cases from stakeholder requests.
- Structure the use cases into use-case diagrams.
- Generate the use-case specifications.
- Review the use-case specifications.
- Analyze the use-case specifications and generate the analysis model.
- Review the analysis model.
- Generate the design model based on the analysis model.

The whole process is divided into two parts based on different concerns. The first part addresses NL requirements analysis and use-case modeling. The second part is concerned with the use-case realization and class model generation. The artifacts and activities in the process are shown in Fig. 1. The output of the requirements phase is a use-case model. Use cases are means to capture the contracts between the stakeholders of a system and its behavior [1]. A use-case model comprises diagrams in UML and specifications that record sequences of actions that a system can perform by interacting with outside actors.

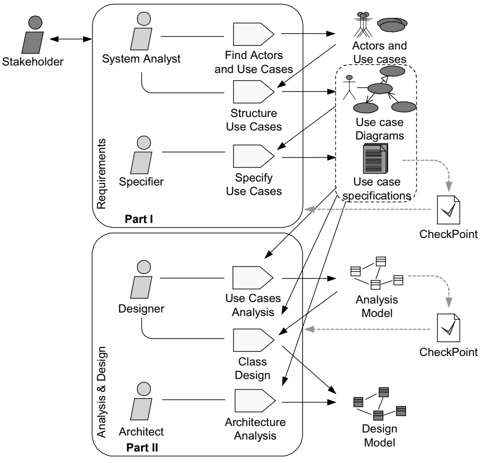


Fig. 1. Developers, activities, and artifacts in OOAD

2.1 NL Processing and Use-Case Modeling

Stakeholders' requests are parsed by a natural language parser to identify use cases. The detailed description of each use case is parsed and analyzed to generate a use-case specification document.

2.1.1 Natural Language Parsing

Stakeholders' requests documented in natural language are analyzed and processed by a parser. A shift-reduce parser is applied in UCDA. The typical structural elements of sentences that are recognized by the parser are listed in Table 1. The presence of coordinated structures and modifiers will increase the complexity of sentences. The tool can identify some complex sentences and break them down into simpler sentences. The rules for sentence reconstruction are listed in Table 2. The parser converts input requests into sentences, and each sentence into an abstract representation of its syntactic structure. For example, "*students request a course catalog*" can be tagged as *['students'/'N', 'request'/'V', 'a'/'ART', 'course'/'N', 'catalog'/'N'].*

Table 1. Sample tag set

Tag	Description	Tag	Description
N	Noun	P	Preposition
V	Verb	DET	Determiner
ART	Article	ADV	Adverb
ADJ	Adjective	CONJ	Conjunction
Q	Quantifier	auxV	Auxiliary Verb

Table 2. Rules for sentence reconstruction

Complex Structure	Simplified Structure
NP1 + Verb1 + NP2 + Verb2 + NP3	(i) NP1 + Verb1 + NP2
	(ii) NP2 + Verb2 + NP3
{Q, ADJ, V-ing, DET} + N	N
auxV + V	V
Sentence1 {AND/OR} Sentence2	(i) Sentence1
	(ii) Sentence2
NP + VP + NP1 and NP2	(i) NP + VP + NP1
	(ii) NP + VP + NP2

NP – Noun Phrase; VP – Verb Phrase

2.1.2 Use Case Identification

The candidate actors are derived from nouns, especially those that are subjects of the statements, and the candidate use cases are derived from the verb phrases acting as actors' predicates. If a candidate actor is not found in the glossary, it will be removed from the candidate actor set. We also apply heuristics to help the developer to distill the candidate sets and identify actors and use cases. Typical heuristics to distill actors are:

1. Who will supply, use, or remove information?
2. Who will use the functionality?
3. Who will support or maintain the system?
4. What are the system's external resources?
5. What are the other systems that are needed?

The heuristics to distill use cases are as follows:

1. For each actor, what are the tasks?
2. Does the actor have to be informed about certain occurrences in the system?

When use cases are identified from the requests, detailed information is needed for each use case to generate the use-case specification. The following is part of the requirements for an ATM (automated teller machine) system.

> The ATM will service one customer at a time. A customer will start a session when s/he inserts an ATM card. The customer will then be able to perform one or more transactions.

The actor identified from this paragraph is "customer", and use cases identified are "start session" and "perform transactions".

2.1.3 Use-Case Specification

A template is used to standardize the use-case specifications. The template contains such entries as use-case name, flow of events, special requirements, preconditions, postconditions and extension points. To enable the automated realization of use-case specifications in NL, we introduce a set of use-case schemas to normalize the NL statements. Table 3 lists the use-case schemas. The structures of simple statements are identified during parsing. They are transitive (a.k.a. monotransitive), intransitive, ditransitive, intensive, complex transitive, prepositional and non-finite [7].

Table 3. Use-case language schemas

Basic	This schema applies to all the events. The sentences in the form of this schema are simple statements.
If-then	This schema is used for alternative flows. An if-clause may consist of one or more condition statements, each of which can be described using the basic schema. A then-clause contains a flow of events.
Do-until	This schema describes a repeated event or sequence of events under a certain condition. A do-clause may contain an event or a sequence of events. An until-clause may consist of one or more condition statements, on which the iteration will stop and the flow goes to next step.
Con-Noc	This schema describes the performance of two or more activities during the same time interval, i.e., concurrency. This schema starts with a Con and ends with a Noc.

2.2 Class Model Generation

2.2.1 The Use-Case Processing Method

To perform use-case driven analysis and design, we propose a use-case processing method as follows:

For each use case,

a. elicit the analysis classes, identify their stereotypes, and generate a robustness diagram;

b. decompose the system’s behavior, distribute the behavior to analysis classes, and generate a collaboration diagram.

For the analysis classes,

a. describe responsibilities;
b. describe associations and establish them in the class diagram;
c. identify the generalization relationships and establish them in the class diagram.

2.2.2 Rules for Class Model Generation

Domain knowledge is very important in object identification and class model generation. The glossary that defines specific terms of the domain represents part of the domain knowledge. If an entity in a use-case specification is found in the glossary, it is a candidate object that may correspond to a class in the class model.

The behavior types of the system, the associations between the stereotype objects, and the structures of the action statements in use-case specifications are associated with each other. Fig. 2 shows a model of the actions in actor-system interaction [1]. Four types of behavior are included in the model. The relationships between the behavior types and the associations between stereotype objects are listed in Table 4. The relationships between the statement structures and the behavior types are summarized, and applied to automate use-case realization. Boundary classes are used to model the system interface that handles the communication between the environment and the system. Entity classes are used to model the real world entities. Control classes are responsible for the flow of events in the use case, and they are application-dependent. Determining the control classes for a problem is very subjective.

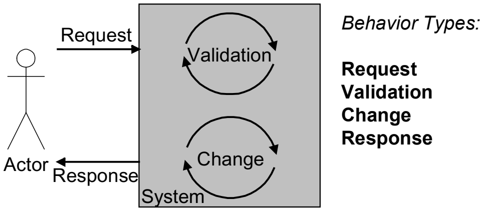


Fig. 2. An actor-system interaction model and 4 behavior types

Table 4. Relationships between behavior types and associations between stereotype objects

Behavior Type	*Association*
Request	웃 —— ⊢◯
Validation	⊢◯ —— ⟳ *and* ⟳ —— ⟳
Change	⟳ —— ◯̲
Response	⟳ —— ⊢◯

웃: actor, ⊢◯: boundary object, ⟳: control object, ◯̲: entity object

We identified the relationships between all statement structures and the behavior types, and represented them in 17 rules for object and message identification [8]. Because of the length of this paper, we only demonstrate a rule for transitive structure shown in Fig. 3, where NP represents noun phrase; VPss represents verb phrase with the statement structure; PP represents prepositional phrase; Vgp represents verb group; and Prep represents preposition [8].

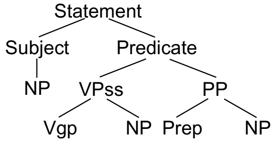


Fig. 3. The structure of a transitive statement

The rule for object identification is:

> **Rule:** *If* the structure of a statement is transitive (shown in Fig. 3), and Subject/NP//Noun(head) is an actor, *then* this statement is corresponding to the Request behavior type and Predicate/PP/NP//Noun(head) is a boundary object if it exists in the glossary, and Predicate/VPss/NP/Noun(head) is an entity object if it exists in the glossary.

If there are two objects or an actor and an object in one statement, an association between them is identified. To generate the collaboration diagram, the messages contained in one scenario are identified. The corresponding rule for message identification is:

> **Rule:** *If* Subject/NP//Noun(head) is an actor, and Predicate/PP/NP//Noun(head) is a boundary object, *then* the action is Predication/VPss/Vgp/Verb(head) + Predication/VPss/Vgp/NP//Noun(head), the sender is Subject/NP//Noun(head) and the receiver is Predicate/PP/NP//Noun(head).

The responsibilities of the classes can be identified from the messages in the collaboration diagrams. Each message consists of a sender, a receiver and an action. The receiver has the responsibility for the execution of the action. The messages in collaboration diagrams are transformed to the classes' responsibilities in this way.

Composition and generalization are two kinds of class relationships to be identified in the class model of the system under development. Some aggregation relationships can be derived from the use-case inclusion relationships.

> **Rule:** *If* one use case includes another use case, *then* a composition relationship is likely to exist between the core control classes identified from the use cases.

Class generalizations can also be identified from use-case generalization relationships.

> **Rule:** *If* one use case has a generalization relationship with another use case, *then* a generalization relationship is likely to also exist between the core control classes identified from the use cases.

2.2.3 Rules for Analysis Model Validation

We propose a method to validate the analysis model, especially the robustness diagrams. There are some constraints for objects and associations in a robustness diagram according to its semantics. The rules listed in Table 5 are derived from the constraints and used for robustness diagram validation.

Table 5. Rules for robustness diagram validation

Case	*Validation*	*Suggestion*
웃—웃	*Not allowed.*	웃—⊢○—↻—⊢○—웃
웃—⊢○	*Allowed*	
웃—↻	*Not allowed.*	웃—⊢○—↻
웃—○̲	*Not allowed.*	웃—⊢○—↻—○̲
⊢○—⊢○	*Not allowed.*	⊢○—↻—⊢○
⊢○—↻	*Allowed.*	
⊢○—○̲	*Not allowed.*	⊢○—↻—○̲
↻—↻	*Allowed.*	
↻—○̲	*Allowed.*	
○̲—○̲	*Not allowed.*	○̲—↻—○̲

웃: actor, ⊢○: boundary object, ↻: control object, ○̲: entity object

3 UCDA: Use-Case Driven Development Assistant

3.1 Overview

To implement the methodology, we develop a CASE (Computer Aided Software Engineering) tool named UCDA (Use-Case driven Development Assistant). Just like the methodology, UCDA is composed of two parts: one part for NL requirements processing and use-case modeling, and the other for use-case realization and class model generation. The architecture of UCDA is shown in Fig. 4. UCDA is integrated seamlessly with IBM Rational Rose. The user can manage UCDA with Rational Rose's Add-in manager. Most artefacts generated by UCDA are represented in XML and visualized in Rose.

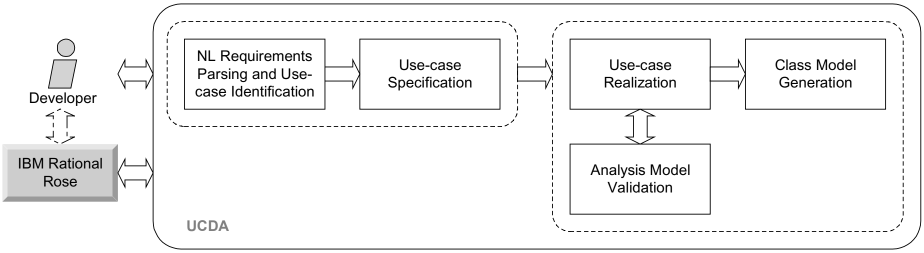


Fig. 4. UCDA architecture

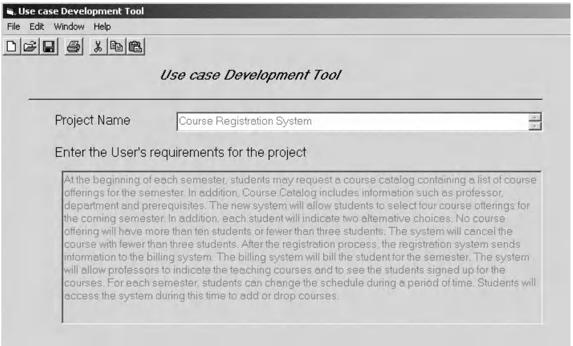
The features of UCDA currently implemented are as follows:

1. Parse the NL requirements and identify actors and use cases, and then generate the use-case diagram in Rational Rose.
2. Assist the user to finish use-case specification.
3. Realize the use cases, identify the classes, and generate robustness diagrams and collaboration diagrams in Rational Rose.
4. Validate the analysis class model via robustness diagrams.
5. Generate the class model in Rational Rose.

Not all the activities can be fully automated especially the use-case specification. The user needs to interact with UCDA to supply the necessary information, and the tool will help the user to develop a model in UML for further revision.

3.2 Use-Case Modeling Environment

When the user has only the requests, s/he can start to analyze with UCDA. Fig. 5 is the environment for requirements parsing. The user needs to paste or edit the requests of a project in it. Then UCDA can help the user identify the use cases from the request and generate the use-case diagram in Rational Rose.



Use case Development Tool

Project Name: Course Registration System

Enter the User's requirements for the project

At the beginning of each semester, students may request a course catalog containing a list of course offerings for the semester. In addition, Course Catalog includes information such as professor, department and prerequisites. The new system will allow students to select four course offerings for the coming semester. In addition, each student will indicate two alternative choices. No course offering will have more than ten students or fewer than three students. The system will cancel the course with fewer than three students. After the registration process, the registration system sends information to the billing system. The billing system will bill the student for the semester. The system will allow professors to indicate the teaching courses and to see the students signed up for the courses. For each semester, students can change the schedule during a period of time. Students will access the system during this time to add or drop courses.

Fig. 5. The environment for NL requirements parsing and use-case identification

Then the user can specify the use cases with the assistance of UCDA. UCDA parses the user's input information and normalizes it based on use-case language schemas. The structure of each statement in the flow of events is identified, and all statements are encoded in XML. An example use-case specification with XML markups removed is as follows. Note that person and number effects on verbs are removed.

Actors: customer, bank
Flow of Events:
Basic Flow:

1. the system start withdrawal transaction;
2. the customer select the account on the customer console;
3. the system get the account from the customer console;
4. the customer select the amount on the customer console;
5. the system get the amount from the customer console;
6. the system generate the withdrawal transaction information;
7. the system send the withdrawal transaction information to the network connection;
8. the bank get the withdrawal transaction information from the network connection;

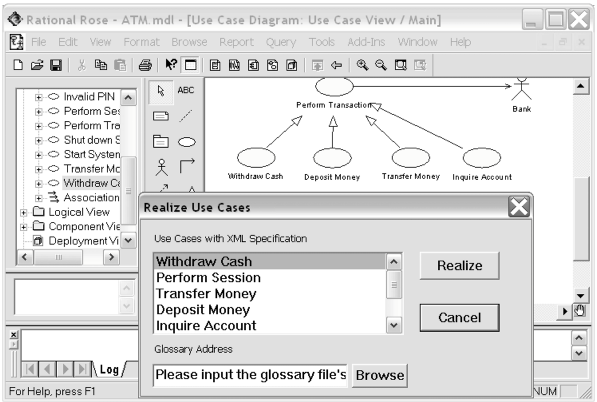
9. the bank send the withdrawal transaction approval to the network connection;
10. the system get the withdrawal transaction approval from the network connection;
11. the system dispense the cash in the cash dispenser;
12. the customer get the cash from the cash dispenser;
13. the system record the withdrawal transaction information into the log;
14. the withdrawal transaction end;

Alternative Flow:

If the bank do not approve the withdrawal transaction,
then i. the system display an error message on the customer console;
ii. the system record the withdrawal transaction information into the log;
iii. the withdrawal transaction end;

3.3 Use-Case Realization Environment

When the use-case model is ready, the user can use UCDA to realize the use cases and generate the class model. The functions of UCDA can be accessed via Rose's menu. All the diagrams generated by the tool are visualized in Rational Rose. The environment for use-case realization is shown in Fig. 6. The user can set the glossary and select a use case to realize. When collaboration diagrams are generated, the tool can distribute the behavior and generate the class model in Rational Rose. The robustness diagram generated by UCDA based on the example specification in Sect. 3.2 is shown in Fig. 7 and the corresponding collaboration diagram is shown in Fig. 8.



Realize Use Cases

Use Cases with XML Specification

Withdraw Cash
Perform Session
Transfer Money
Deposit Money
Inquire Account

Glossary Address

Please input the glossary file's

Fig. 6. The environment for use-case realization cooperating with Rational Rose

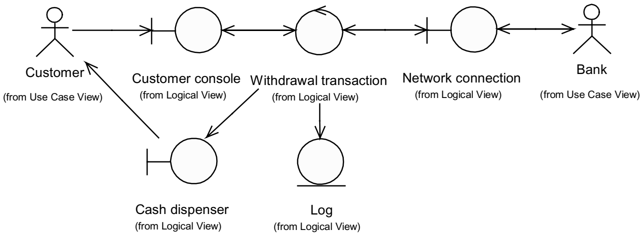


Fig. 7. A robustness diagram generated by UCDA

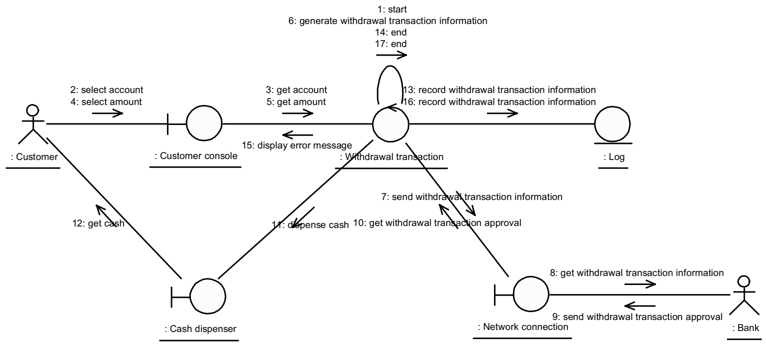

1: start
6: generate withdrawal transaction information
14: end
17: end
2: select account
4: select amount
3: get account
5: get amount
13: record withdrawal transaction information
16: record withdrawal transaction information
15: display error message
: Customer
: Customer console
: Withdrawal transaction
: Log
7: send withdrawal transaction information
12: get cash
11: dispense cash
10: get withdrawal transaction approval
8: get withdrawal transaction information
9: send withdrawal transaction approval
: Cash dispenser
: Network connection
: Bank

Fig. 8. A collaboration diagram generated by UCDA

4 Conclusion

A methodology for natural language requirements analysis, use-case modeling and use-case driven analysis and design is presented. The methodology comprises good practices from both natural language requirements analysis and the use-case driven analysis and design. Use-case language schemas are proposed to normalize use-case specifications, and the methods to automate object identification and class model generation based on statement structures are discussed. A CASE tool was developed to support the methodology. A future research topic is to implement features of software architecture analysis and integrate it with the UCDA and Rational Rose.

References

1. Cockburn, A,: Writing effective use cases. Addison-Wesley (2000)
2. Bois, P. D., Dubois, E., Zeippen, J.M.: On the use of a formal RE language-the generalized railroad crossing problem. Proceedings of the Third IEEE International Symposium on Requirements Engineering, Annapolis MD (1997) 128-137
3. Li, X., Liu, Z., He, J.: Formal and use-case driven requirement analysis in UML. 25th Annual International Computer Software and Applications Conference, COMPSAC2001 Chicago (2001) 215-224
4. Boyd, N.: Using natural language in software development. Journal of Object-Oriented Programming 11(9) (1999) 45-55
5. Overmyer, Scott P., Lavoie, B., Rambow, O.: Conceptual modeling through linguistic analysis using LIDA. Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, Toronto (2001) 401-410
6. Niba, L.C.: The NIBA workflow: From textual requirements specifications to UML-schemata. Proceedings of International Conference on Software & Systems Engineering and their Applications, ICSSEA 2002, Paris (2002)
7. Roberts, P.: Patterns of English. Harcourt, Brace and Company (1956)
8. Liu, D.: Automating transition from use cases to class model. MSc Thesis, University of Calgary, Calgary (2003)

Using Cognitive Modelling Simulations for User Interface Design Decisions

Bruno Emond[1] and Robert L. West[2]

[1] Institute for Information Technology, National Research Council Canada,
1200 Montreal Road, Building M-50, Ottawa, ON, Canada. K1A 0R6
bruno.emond@nrc-cnrc.gc.ca

[2] Department of Psychology, and Department of Cognitive Science, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada. K1S 5B6
robert_west@carleton.ca

Abstract. This paper argues for the relevance of cognitive modelling and cognitive architectures to support user interface design decisions. From a human-computer interaction point of view, cognitive modelling can have benefits both for theory and model building, and for the design and evaluation of systems usability. Cognitive modelling research applied to human-computer interaction has two complimentary objectives: 1) to develop theories and computational models of human interactive behaviour with information technologies, and 2) to use the computational models as building blocks for the design, implementation, and evaluation of interactive technologies. As an example of application of cognitive modelling to technology design, the paper presents a simulation of interactive behaviour with five different adaptive menu algorithms: random, fixed, stacked, frequency based, and activation based. Results of the simulation indicate that fixed menu positions have an advantage over adaptive menus in taking advantage of the capability of human memory in human-computer systems.

1 Introduction

The idea of using user models as a tool to understand and design human computer environment has been pursued for some time under the discipline of cognitive engineering [1, 2]. Applications of cognitive engineering aim mostly at modelling, predicting and evaluating human performances in computer environments. Recently, cognitive engineering applications have focused on high-fidelity simulations so that cognitive models can interact directly with a software application, modelling visual and auditory perception as well as motor actions [3,4]. The other focus is on anchoring cognitive simulations in cognitive architectures, which provide relatively complete proposals about the structure of human cognition.

Computer simulations and cognitive modelling based on cognitive architectures can be an important methodological component in the study and design of interactive technology. From a human-computer interaction point of view, cognitive modelling

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 305–314, 2004.

can have benefits both for theory and model building, and for the design and evaluation of systems usability. Cognitive modelling research applied to human-computer interaction has two complimentary objectives: 1) to develop theories and computational models of human interactive behaviour with information technologies, and 2) to use the computational models as building blocks for the design, implementation, and evaluation of interactive technologies. The development of cognitive modelling simulation techniques could lead to important applications in psychology, and software engineering [5].

This paper is divided in five sections. Section 2 will present some general ideas about cognitive modelling methodology. Section 3 will outline some benefits of using low fidelity prototyping in combination with cognitive models. The current approach is essentially focused on the task structure rather than the interface layout. Finally, section 4 will present the results of a simulation of human computer interaction with four types of adaptive menus: interactive behaviour with five different adaptive menu algorithms: random, fixed, stacked, frequency based, and activation based. The simulation contains also an evaluation of the effect of cognitive strategies used to interact with menus. Section 5 presents a brief conclusion.

2 Cognitive Modelling Methodology

Cognitive modelling has its roots in cognitive architecture research and unified theories of cognition [6,7]. Cognitive architectures are relatively complete proposals about the structure of human cognition. A cognitive architecture provides the resources for developing models. These resources take the form of a set of specifications regarding the functional invariants [8] related to knowledge representation, knowledge processing, memory, perception, and motor actions. Some examples of cognitive architectures are SOAR [6], ACT-R [9], EPIC [3], and CI [10]. Each of these architectures has its strength and was initially developed with some intended modelling purpose. ACT-R is mainly focused on problem solving and memory, SOAR on problem solving and learning, EPIC on multiple task performance, and CI on text comprehension.

Anderson [7, 9] presents an interesting way to understand the place of cognitive architectures and cognitive models in the scientific investigation of cognition. According to Anderson, a framework is a general concept for understanding a domain, but it does not have predictive power, whereas a theory is a precise system that embodies specific framework level concepts and can be used to make predictions. For example, the idea that cognition can be understood using production rules (i.e., if/then rules) is a framework level assumption, while specific implementations of production rules to do so constitutes a theory. Cognitive architectures, therefore, can be considered theories capable of explaining complex cognitive behaviours. Models are the result of applying a theory to a specific task or phenomenon to predict measures of performance such as processing time, errors, learning rates and learning patterns.

The cognitive modelling methodology is mostly an iterative methodology similar to the learning cycle in HCI research [12] that goes through successive cycles of theory building, computational artifact construction, and empirical evaluation. Figure 1 presents the process of cognitive modelling [13]. One can see from this figure that empirical data plays a crucial role in cognitive modelling. It is certain that simulations, as a method a scientific inquiry, can best be advanced and tested by the concurrent and complementary use of empirical methods [11].

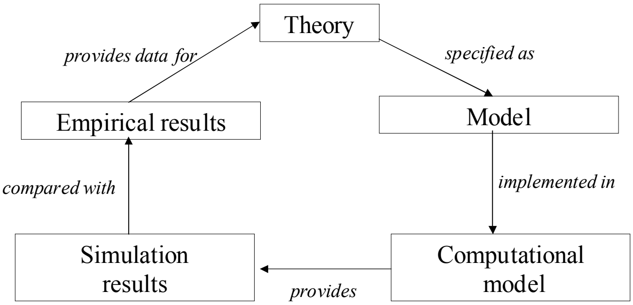


Fig. 1. Cognitive modelling process

Another point of view on cognitive modelling methodology is the cognition-artifact-task triad [14] for understanding what is meant by interactive behaviour. The interactive behaviour for any given artifact-task combination arises from the limits, mutual constraints, and interactions between and among each member of the cognition-artifact-task triad. The cognitive modelling approach considers that all three components must be taken into account [4].

3 Cognitive Modelling Using Low Fidelity Prototypes for Usability Testing

Computer simulations based on cognitive architectures can be an important methodological component in the study and design of interactive technology. According to Ritter [15], three elements are needed for modelling human-computer interaction tasks: 1) a cognitive model that simulates the cognitive performance of a human performing a task, 2) a task simulation that provides the task as well as the interface that will be used by the cognitive model, and 3) a linkage mechanism that simulates human perception and action so that the cognitive model can communicate with the task simulation.

These elements are present in cognitive modelling environments such as ACT-R/PM [4]. The set of tools available to model interactive behaviour range from indirect interaction with software applications through abstract specifications of a user interface [16,17] through application mock-up (ACT-R/PM), or direct interaction with software applications [15,18]. Our research has focused mainly on indirect interaction with abstract specifications of a user interface for the purpose of usability testing in the context of rapid prototyping.

The prototypes used in usability testing can also range from high fidelity to low fidelity. High fidelity prototypes are fully functional or almost fully functional interfaces. They have the advantage of providing human subjects with a realistic experience but have the disadvantage of being time consuming to develop. Low fidelity prototypes are mock-ups that have limited functionality. The advantage of low fidelity prototypes is that they are fast and cheap, and therefore very useful for testing potential interface designs early in the design process. Currently there are several projects to get simulated users to interact with relatively high fidelity interfaces [4]. The idea is to develop systems that allow simulated users to interact with the same software that human subjects interact with. This approach is good for high fidelity prototyping but may not be the best choice for low fidelity prototyping.

Because we are interested in testing simulated users early on in the rapid prototyping process, we have focused on low fidelity prototyping. To be useful, low fidelity prototyping systems must be relatively quick and easy to use. They also need to capture the elements of the full interface design that drive the way that human users interact with it. If completely successful in this regard, a low fidelity prototype is just as effective as a high fidelity prototype for testing human subjects. However, there is no way to know if you have succeeded without also building a high fidelity prototype and testing to see if people behave the same way with it. The same is not true for simulated users. Since we know how a simulated user works, we can know what aspects of an interface prototype will affect it and what will not. Also, we know that a simulated user will not be affected by the realism of the experience. In fact, a simulated user does not require a visual interface at all. It just needs to be told what is there and what are the effects of its actions.

ACT-R/SOS (Simple Operating System for ACT-R) is an application that we have developed to implement both the environment simulation and the linkage mechanism [16,17] between an ACT-R cognitive model and the simulated environment. The immediate purpose of SOS is to provide support low fidelity simulations in a rapid prototyping environment [16,17]. SOS interacts with a formal specification of the interface created within the SOS system (i.e., a simulation model of the actual interface). This allows interface designs to be tested against ACT-R agents.

4 Usability Testing of Adaptive User Interfaces with Simulated Users

The intention of this section is to show that cognitive models, anchored in unified theory of cognition, support the production of specific predictions, and the clarification of empirical questions to be addressed prior to data collection. As an analogy, transportation simulations support hypothesis testing and infrastructure design without having to build physical infrastructures to know their adequacy. The development of cognitive modelling simulation techniques could lead to important applications in psychology, and software engineering [5]. As an example of the application of cognitive modelling to technology design, this section presents a simulation of interactive behaviour with five different adaptive menu algorithms: random, fixed, stacked, frequency based, and activation based. The results of the simulation can be used to guide both initial technology and empirical study design decisions, in the context of assumptions about user cognitive processing.

The purpose of the simulation is to explore the possible consequences on human performance for a variety of menu generation algorithms. The menus are generated as the product of the user interaction with a set of data. In this case, a target is presented to the cognitive model, which must select the corresponding menu-item in a menu. When the target is presented, the simulated user tries to recall its position in the menu and then proceeds to select the corresponding menu item. One can think that an application, such as email, could have a similar functionality. A message comes in and the user classifies the message by choosing the appropriate menu item. Initially the menu is empty, but as messages arrive, categories are created to classify the messages up to a point of forming a finite list of categories or topic. Five menu generation algorithms were tested:

Random positions menu: As new targets are presented, they are simply assigned a random position in the list. The menu is composed randomly each time it is accessed. This algorithm is similar to the random menu selection task [4]. The classification of a target to the corresponding menu item does not affect the random process.

Fixed positions menu: As new targets are presented, they are simply assigned a position at the end of the list. As long as known targets are present, no changes occur to the menu. The classification of a target to the corresponding menu does not affect the positions of items in the menu.

Stack menu: As new targets are presented, they are simply assigned a position at the beginning of the list. As long as known targets are present, no changes occur to the menu. The classification of a target moves the corresponding menu item to the beginning of the list and pushes the other items down.

Frequency menu: As targets are presented, they are assigned a position in the list that corresponds to its frequency therefore a new target would be placed at the end. The classification of a target might not change the menu if the addition does not change the relative frequencies of menu items. The menu items are sorted from the most to the least frequent.

Activation menu: As targets are presented, they are assigned a position in the list that corresponds to their activations (see below for a definition of activations). The classification of a target might not change the menu if the addition does not change the relative activations of menu items. The menu items are sorted from the most to the least active. This menu generation algorithm aims at mirroring the memory of the cognitive model for the menu items.

The present implementation of the concept of activation finds its origin in the ACT-R cognitive architecture [9]. In this cognitive architecture, activation is a measure of the degree that past experiences indicate that they will be useful at a particular moment in time. The base-level activation of a memory chunk represents how recently and frequently it is accessed. The base-level activation therefore takes into account both the number of times a memory chunk is accessed as well as the amount of time since it was last accessed. This is an important feature of activation because its value is not constantly growing as the number of references increases, but is subject to decay across time. The role of decay rate is an important factor in models of human memory. The formula to determine the base-level of activation of a chunk of memory at time t after its creation, taking into account all intervening references or accesses to the chunk, is the following:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) + \beta \tag{1}$$

Where β is the expected activation value at the creation time of a chunk in memory, d is the decay rate, and t_j are the times it has been accessed. In the case where a chunk has been accessed only once since its creation time, a simplified version of the formula would be the following:

$$B(t) = \beta - d\ln(t) \tag{2}$$

The parameter d is the decay rate and can be set to any values between 0 and 1, while $\ln(t)$ is the logarithm of the time t. Anderson and Schooler [19] have shown that the log odds of something reoccurring in natural settings approximates this function. Thus activation level can be viewed as representing the probability of events reoccurring in the environment.

The cognitive model that interacts with the adaptive menu systems is composed of a set of productions rules that control the action of the simulated users. Also the simulated users have a memory buffer holding the memory chunks with their specific levels of activation. The productions rules implement the menu searching strategy. Empirical evidence from collecting data on eye movements on random menu selection tasks indicates that the strategy used by people tends to be either top down or random search. The general result of study of selection of random menu selection is that the response time is a function of the target location, with menu items located at a lower position in the menu generating longer response times [4].

These strategies, however, are likely to be dependent on random nature of the task. Menus with different structures could elicit different strategies. The cognitive models used in the simulation varied in terms of strategies that could be pursued. Two general strategies were implemented in the cognitive simulations. An initial strategy used by a simulated user could be either to retrieve from memory the location of a menu item or to scan the menu to find the menu item without a prior recall of its anticipated position. A second strategy was to scan the menu either from the top or from a random position when a menu item has not been found. The options of this second strategy could be pursued when memory is not used prior to search the menu, or when a retrieval failure occurs, or after an initial lookup is not successful. The two strategies can be combined and give four possible combinations. These combinations are:

A) Search first (do not retrieve) and then scan from an initial random position;

B) Search first (do not retrieve) and then scan from the top of the menu;

C) Retrieve first, look at the anticipated position (if retrieval is successful) and then scan from a random position (if the first lookup is not successful);

D) Retrieve first, look at the anticipated position (if retrieval is successful) and then scan from the top of the menu (if the first lookup is not successful)

For each of these four conditions, a cognitive model has to process ten successive sets of thirty targets in the five adaptive menu conditions. The maximum size of a menu is of twelve menu items. The targets submitted to the simulated user were composed of a set of target labels that were presented sequentially. The global list of targets (300 targets) was composed of ten subsets of thirty targets. These subsets were identical in terms of the elements they contained but not necessarily in terms of their sequential positions. The subsets were randomly constructed with a list of targets with different frequencies appearing early (positions 1 to 15), late (positions 16 to 30), or randomly (positions 1 to 30). The following target labels were used to compose the data set:

Random targets (1-30):	T08 T08 T08 T08 T11 T11 T11 T02 T02 T05
Early targets 1-15:	T07 T07 T07 T07 T10 T10 T10 T01 T01 T04
Late targets 16-30:	T09 T09 T09 T09 T12 T12 T12 T03 T03 T06

A complete factorial analysis of variance performed on the simulation output showed that all terms were significant. The main effects were: first strategy [S1: search / retrieve] (F(1,160)=953.81 MS=92.175 p<0,0001), second strategy [S2: from top / random position] (F(1,160)=9.63 MS=0.931 p<0,005), and menu types [Menu: random / activation / frequency / stacked / fixed] (F(4,160)=553.22 MS=53.462 p<0,0001). Two way interactions were: S1*S2 (F(1,160)=263.56 MS=25.470 p<0,0001), S1*Menu (F(4,160)=664.04 MS=64.172 p<0,0001), S2*Menu (F(4,160)=22.66 MS=2.190 p<0, 0001). Finally, the three way interactions was also significant (F(4,160)=33.41 MS=3.228 p<0,0001). The average number of scans per cycle of 30 targets was used for the analysis. The first learning cycle of 30 targets, which were initially learned by the simulated users, were removed from the analysis. The data consisted of 180 observations composed of the average scans over 30 simulated users for 9 learning cycles in each of the 20 cells of the design.

Figure 2 plots the results for the three way interactions. From Figure 2, one can see that the strategy of scanning a menu without doing a prior retrieval, coupled with the strategy of scanning from the top position produced the least number of scans both for the activation and frequency based menus. On the other hand, the strategies of retrieving a menu item position and using a random scan position resulted in the least number of menu items scanned for the stacked and fixed menus. Although, it is important to note that the data is about the number of scans before a successful match is found.

As Figure 3 shows, the simulated users learned all the menu item positions for the fixed menu condition by the seventh learning cycle, which did not cause any additional menu item to be scanned. Figure 3 displays the learning history for the strategies of retrieving first and scanning from the top in case of a failure to find a menu item. The next best menu option in this condition is the stacked menu that shows a tendency to approximate the performance on the fixed menu. Finally, the frequency and the activation menu options support the lowest cognitive model performance, even worst than a random menu for these strategies (retrieve first, search from top).

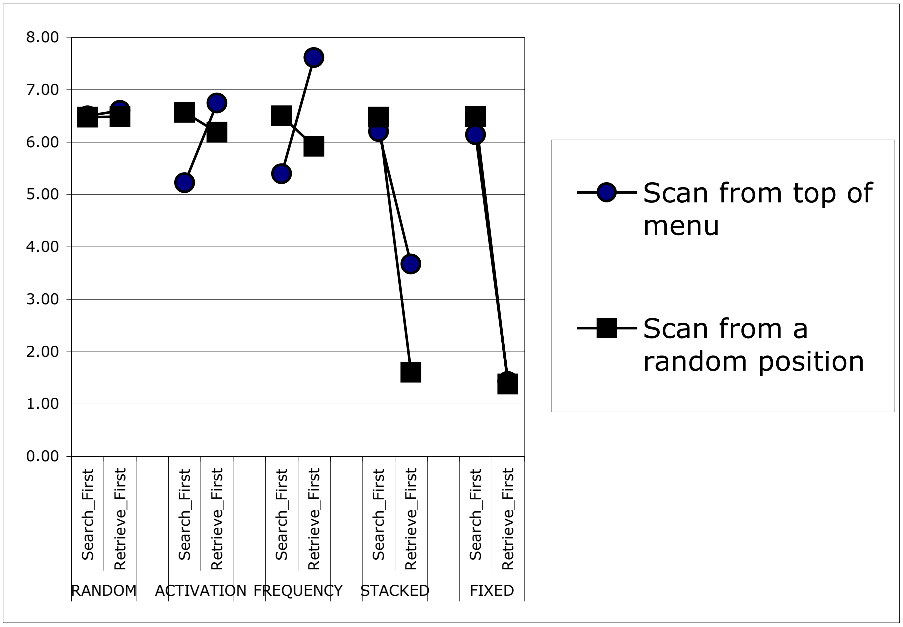


Fig. 2. Average number of menu item scanned as a function of strategies and adaptive menu options

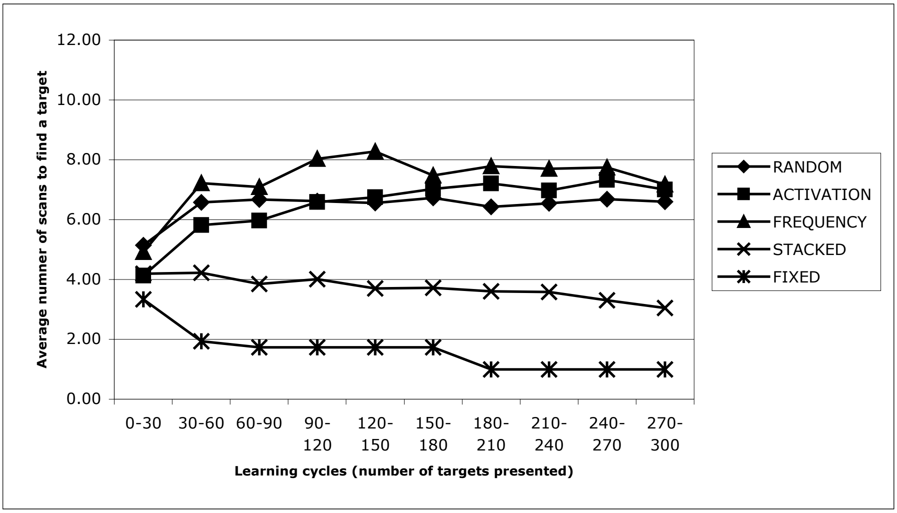


Fig. 3. Average number of menu item scanned as a function of learning cycles and adaptive menu options (Strategies: retrieve first, search from top of menu).

The results of this simulation indicate that there is an interaction between adaptive menus and cognitive strategies used to access them. The results suggest that adaptive menus placing emphasis on access frequency (frequency and activation) would tend to support scanning strategies because the gain in scanning costs related to trying to retrieve from memory the position of a menu item. On the other hand, menus that are stable (stacked and fixed) would tend to promote the use of retrieval strategies because of the reduction of menu scanning costs. Overall though, the performance of simulated users across all strategies and menu conditions indicate that stable menus (stacked and fixed) seem to offer the best support when the user has an immediate access to a list of menu items. Because adaptive menus are constantly changing, the human-computer system cannot take advantage the human capacity to memorized menu item positions, which results in many retrieval and matching failures, and as a consequence, menu item scans.

5 Conclusion

The above simulation is a good example of how cognitive modelling can be used to predict human performance on human-computer interaction tasks. From the simulation results, it is possible to draw the conclusion that a stable interface provides a better support for users' memory than an interface that aims at mirroring the distribution of events in the environment and/or users' memory of these events. This simulation could be the basis for formulating specific hypothesis, which could be tested empirically. More simulations are also required to explore of the effect of no-retrieval strategy to see how adaptive menus support human performances. For example, the situation is different for menus in a menu bar, which require pointing to the first position in order to see the list. The current simulation is not completely accurate in this respect because a simulated user can access directly a menu item without having to first locate the first menu item (or menu title). Studies of fixed menu positions, show that even with well-memorized menu positions, there is an increase of time to access menu items located at the bottom of the menu item list [20]. We believe that usability testing can be significantly improved through the use of simulated users. However, we do not believe that simulated users should be used to replace human usability testing. Rather, we suggest that the two techniques complement each other.

References

1. Card, S. K., Moran, T. P., and Newell, A. (1983). The Psychology of Human-Computer Interaction , Lawrence Erlbaum.
2. Norman D.A. (1986). Cognitive Engineering. In Norman & Draper (eds) User Centered Systems Design. Hillsdale NJ. Erlbaum.
3. Kieras, D.E., Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction, 12, 391-438.

4. Byrne, M. D., (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. International Journal of Human-Computer Studies, 55, 41-84.
5. Ritter, F. E., (ed.) (2001). Special issue on using cognitive models to improve interface design. International Journal of Human-Computer Studies, 55:1-14.
6. Newell, A. (1990). Unified theories of Cognition. Cambridge, MA: Harvard University Press.
7. Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.
8. Pylyshyn, Z.W. (1984). Computation and cognition: toward a foundation for cognitive science. MIT Press: Cambridge, MA.
9. Anderson, J. R., Lebiere, C. (1998). The Atomic Components of Thought. Mahwah, NJ: Lawrence Erlbaum Associates.
10. Kintsch, W. (1998). Comprehension. Cambridge: Cambridge University press.
11. Goldspink, C. (2002). Methodological implications of complex systems approaches to sociality: simulation as a foundation for knowledge. Journal of Artificial Societies and Social Simulation, 5(1), http://jasss.soc.surrey.ac.uk/5/1/3.html.
12. Olson, G.M., Olson, J.S. (1997). Research on computer supported cooperative work. In M. Helander (Ed.), Handbook of Human computer interaction, Amsterdam : Elsevier, 1433- 1457.
13. Dijkstra, A.F.J., De Smedt, K.J.M.J. (1996). Computer models in psycholinguistics. In A.F.J. Dijkstra, K.J.M.J. De Smedt (Eds.), Computational psycholinguistics: AI and connectionist models of human language processing, London: Taylor & Francis, 3-23.
14. Gray, W. D., Altmann, E. M. (2001). Cognitive modeling and human-computer interaction. In W. Karwowski (Ed.), International encyclopedia of ergonomics and human factors, 1, 387-391.
15. Ritter, F.E., Baxter, G.D., Jones, G., Young, R.M. (2000). Supporting cognitive models as users. ACM transactions on computer-human interaction, 7(2):141-173.
16. West, R.L., Emond, B. (2001). Can cognitive modeling improve usability testing and rapid prototyping ? Fourth International Conference on Cognitive Modeling, George Mason University, Fairfax, Virginia, USA, July 26 - 28.
17. Emond, B., West, R.L. (2003). Exploring the usability of adaptive menus with a simple object system. Eight Annual ACT-R Workshop Pittsburg, PA : Department of Psychology, Carnegie Mellon University.
18. Zettlemoyer, L.S., St.-Amant, R. (1999). A visual medium for programmatic control of interactive applications. Human factors in computing systems: proceedings of CHI 99, Reading, MA: Addison-Weley, 199-206.
19. Anderson, J.R., Schooler, L.J. (1991). Reflections of the environment in memory. Psychological Science, 2:396-408.
20. Hornof, A.J., Kieras, D.E. (1999). Cognitive modelling demonstrates how people use anticipated location of menu items, ACM CHI 99 proceedings.

An Intelligent Interface for Customer Behaviour Analysis from Interaction Activities in Electronic Commerce

Chien-Chang Hsu and Chang-Wen Deng

Department of Computer Science and Information Engineering,
Fu-Jen Catholic University
510 Chung Cheng Rd., Hsinchuang, Taipei, TAIWAN 242

Abstract. This work proposes an intelligent interface for customer behavior analysis in electronic commerce. The intelligent interface contains three modules, namely, the task editor, action supervisor, and behavior analyzer. The task editor provides an intelligent interface for the system administrator to define the business tasks and domain ontology. The action supervisor is defined as the information collector for monitoring the customer operations, excluding unnecessary operations, and recognizing the behavior patterns. The action supervisor uses the interaction message to extract customer operations, the Bayesian belief network to filter out redundant and irrelevant operations, and the RBF neural networks to recognize the behavior patterns. Finally, the behavior analyzer generates the customer behavior analysis information by measuring the behavior patterns, constructing the personalized domain ontology, and evaluating skill proficiency of the customer.

1 Introduction

Electronic commerce (EC) is the business activity that occurs over the electronic network. The pervasive connectivity of the Internet provides the best mediation for the users, that is, the manufacturer, the broker, the retailer, and the end user, to sell or purchase goods or services. They usually exchange the merchandise through the business platform of the EC systems. The EC business was grouped into three basic models, that is, business-to-business, business-to-consumer, and consumer-to-consumer [12]. The personalized service is the important factor for the above model to attract their customers. The personalized service provides a friendly environment for the customer to buy goods according to the past consuming behaviors. Customer behavior analysis becomes the important function in the system. The customer analysis provides the place to analyze the customer's information as well as help the customer to do business transactions.

Many systems have been proposed for applying the past user transaction history to different applications [6], [7], [8], [11]. These systems are insufficiently specific regarding the analysis of the customer behavior. The system also did not provide the interaction activities analysis for effective customer behavior analysis. Some interface usability tools or systems [1], [2], [4], [5], [9] were designed to overcome the above shortcomings. Many problems still need to be solved. First, most systems use a resident monitoring program in the application system to collect user

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 315-324, 2004.

information from the interaction information between the end user and application system. The monitoring of user behaviors usually uses the cookies and the log files to record the user operations [4], [9]. Cookies were used to keep the user information in the client side. The drawbacks of cookies contain the limited amount of the information in the client side and the dangerous of information loss in the business session. Moreover, the drawbacks of log files include the collection of passive user request information and the complexity of interaction events. Second, most systems acquire the static and specific business information from the content of the web pages. They do not analyze the interaction operations of the user in the applications. Third, most systems assume the user is an experimented operator and may not commit wrong operations. The collected data are presumed correct and valid information. Finally, most systems lack the evaluation of user domain knowledge and skill proficiency in the application domain.

This work proposes an intelligent interface for customer behavior analysis in electronic commerce. The intelligent interface contains three modules, namely, the task editor, action supervisor, and behavior analyzer. The task editor provides an intelligent interface for the system administrator to define the business tasks and domain ontology. The action supervisor is defined as the information collector for monitoring the customer operations, excluding unnecessary operations, and recognizing the behavior patterns. The action supervisor uses the interaction message to extract customer operations, the Bayesian belief network to filter out redundant and irrelevant operations, and the RBF (Radial Basis Function) neural networks to recognize the behavior patterns. Finally, the behavior analyzer generates the customer behavior analysis information by measuring the behavior patterns, constructing the personalized domain ontology, and evaluating skill proficiency of the customer.

The rest of paper is organized as follows. Section 2 introduces the architecture of intelligent interface for customer behavior analysis. Moreover, section 3, 4, and 5 explore the task editor, action supervisor, and behavior analyzer. Subsequently, section 6 demonstrates the application of the intelligent interface in electronic commerce. Finally, section 7 concludes the work.

2 System Architecture

Figure 1 illustrates the architecture of the intelligent interface for customer behavior analysis. The architecture contains three modules, namely, the task editor, action supervisor, and behavior analyzer. The *task editor* provides an intelligent interface for the system administrator to define the business tasks and domain ontology. Specifically, the task editor provides a graphical user interface for the system administrator to define the business tasks, transaction behaviors, and interaction operations. The business tasks describe the goals and intensions that the customers wish to perform. The transaction behaviors describe the behaviors of the customers in accomplishing the tasks. Moreover, the interactions contain the interaction message of the customer, that is, customer events in the browsers and requested information in the HTTP (Hypertext Transfer Protocol). Moreover, the task editor also provides an editing environment for the system administrator to construct the domain ontology.

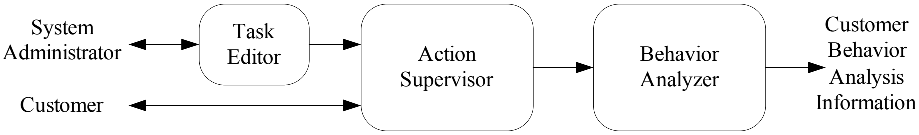


Fig. 1. System architecture

The *action supervisor* monitors the interactions of the customer in application systems. Specifically, the action supervisor collects the interactions performed by the customer in the client-side and server-side. First, the action supervisor captures the customer interactions from the browser and HTTP. The action supervisor then filters the customer operations from the customer interactions and recognizes the behavior patterns based on the business tasks. Finally, the action supervisor forwards the behavior patterns and the domain ontology for use by the behavior analyzer in customer behavior analysis.

The *behavior analyzer* uses the behavior patterns to analyze the activities of the customer. First, it analyzes the behaviors and tasks based on the customer operations. It uses the frequency of behavior patterns to compute the proximity between them. It then evaluates personal domain ontology and skill proficiency. The personal domain ontology stores the terminology and the relations between them of the application domain which was visited or used by the customer. The skill proficiency judges the profile which the customer generally uses for task accomplishment. Finally, the behavior analyzer generates the above behavior analysis information of the customer.

3 Task Editor

The task editor provides a visualized editing environment for the system administrator to create the business tasks and to edit the domain ontology. The business tasks describe the goals that the customers wish to perform. The business tasks contain the tasks involved in carrying out the business process, the behavior patterns and its relations, and the interaction operations. Moreover, the behavior patterns and its relations record the transaction actions related to accomplishing the tasks and the causality relationships between them. The interaction operations contain the customer events in the browsers and requested information in the HTTP. The task editor contains two components, namely, editing tools and editing frame. The editing tool provides six functions for editing the business tasks and domain ontology, that is, task insertion, task deletion, class insertion, class deletion, and task query. The task insertion and deletion provide the function for adding or removing the number of task in the customer interaction. Moreover, the class insertion and deletion provide the function for insertion or deletion the number of class in the framework of the EC ontology. The task query supports the capability to inquire regarding the contents of the specific tasks. The editing frame displays the components in the business tasks and domain ontology.

The menu of task editor contains two functions, that is, type selection and editing tool (Fig. 2). The type selection provides two editing modes for the system administrator to select the desired editing component, that is, business task and domain ontology. The editing tool provides six editing functions for the system administrator as described above. Notably, the hierarchy of the business task contains three levels, that is, business task level, transaction behavior level, and interaction operation level [3].

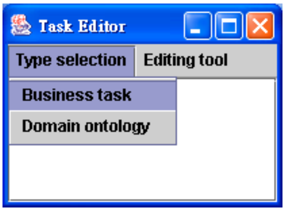


Fig. 2. Task editor

4 Action Supervisor

The action supervisor conducts three tasks, namely, customer operation monitoring, unnecessary operation exclusion, and behavior pattern recognition. The customer operation monitoring monitors customer operations by extracting customer operations from the interaction message between the customer and the system. The customer sends the interaction message by clicking the mouse or pressing the keyboard. The action supervisor uses the delimitation operations, such as, pressing enter on the keyword or single clicking the hyperlink or button, to parse the message into interaction operations. Each interaction operation contains information regarding the interaction events and the HTTP message. The action supervisor uses the web page scripts and applet to capture the interaction events in each session. The interaction events contain the customer events in the client, such as, form submission, button clicking, text input, item selection, hyperlink clicking, and focus objects. Moreover, the action supervisor also captures the HTTP message from the web server. The HTTP message is the request method, path information, query items, protocol, and host name. For example, the customer sends the request message, contained in the header of the HTTP, to the web server. The HTTP message of the request information is the “Get”, “/somedire/page.asp”, “Action=aaa”, “xxx=xxx”, “HTTP/1.0”, and “Host=intsys.csie.fju.edu.tw”.

The exclusion of unnecessary operations eliminates unnecessary customer operations. It uses the Bayesian belief networks to exclude redundant and irrelevant operations. Redundant operations are those which the customer use the same operation for the same subtask. Moreover, the irrelevant operations are those that are not effective in performing the subtask. The directed acyclic graph is used to represent the relationship among customer operations (Fig. 3). The circular node represents the operation of the customer and the link represents the relationship between operations. Both of two operations <O_1, O_2> and <O_3, O_4> can use to do the same subtask ST_1. If the customer operates sequence is O_1, O_3, O_2 and O_5. The operation O_5 is the irrelevant operation and O_3 is the redundant operation for the subtask ST_1. The probability of the subtask ST_1, $P(ST_1)=P(O_1)*P(O_2)$. Redundant and unnecessary operations can be removed by computing the joint probability of the operations according to the causal relations of the Bayesian belief network.

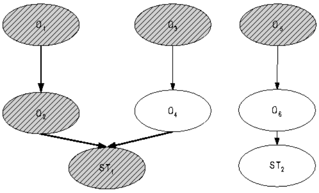


Fig. 3. Bayesian belief network

The behavior pattern recognition uses the RBF neural model as the behavior pattern classifier for discriminating the customer behavior from the interaction message. The RBF neural model contains three layers, namely, the input, hidden, and output layers. Notably, the input layer uses 35 nodes to represent the interaction information. The input of the interaction information includes the interaction events, focus object, request method, and query items. The node number of the interaction information in the input layer is 9, 10, 6, and 10 correspondingly. Each interaction event uses three digits to represent the operation. The character of the focus object, request method, and query items are encoded following the sequence of the alphabetically. Each character then is normalized into a real number ranging between 0 and 1. The node of output layer represents the transaction behaviors. The number of node in the output layer varies according to the behavior patterns in the business tasks.

5 Behavior Analyzer

The analysis of customer behavior is supported by two main processes, namely, the behavior analysis and knowledge and skill proficiency evaluation. The behavior analysis uses the task network to represent the behaviors and behavior correlation of the customer in completing the task. Fig. 4 illustrates the topology of the task network. The square box is used to represent the business task. The circular node is used to represent the name and frequency of the behavior pattern with the focus object. Moreover, the decorated link is used to represent the successive relations between behavior patterns and their frequency. The task network was updated after the customer completed the business task. The behavior analyzer then re-evaluates the content of the task network, that is, the behavior pattern, the relationship of behavior pattern, and the task sequence. Moreover, the relationship of the behavior pattern represents the correlation between behavior patterns. The fuzzy behavior proximity (FBP) is used to measure the correlation,

$$FBP_{i,j}(k+1)=FBP_{i,j}(k)+\alpha(\left\|A_{i,j}(k+1)\right\|-\left\|A_{i,j}(k)\right\|) \tag{1}$$

where i and j is the ith and jth behavior pattern and i is the precedence of j, k is the kth business task, α is the learning rate, and ||Ai,j|| is the frequency of the behavior j follow the behavior i.

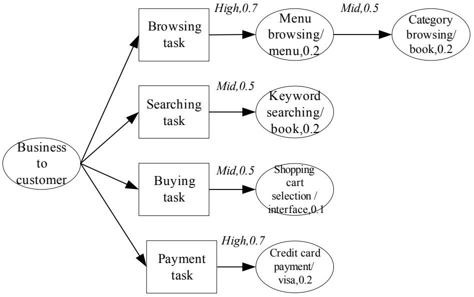


Fig.4. Task network

The knowledge and skill proficiency evaluation computes the proficiency degree of customer. The knowledge proficiency evaluates the personal ontology of the customer. It constructs the personal ontology by mapping the domain terms from the interaction message, that is, focus objects and query items, to the domain ontology. The knowledge proficiency evaluation then uses the explanation-based-learning (EBL) [10] to conduct concept abstraction by providing the domain ontology. The domain ontology is used to build the personal ontology by giving the learning instance, that is, focus object and query items.

The skill proficiency measures the customer proficiency by evaluating the usefulness, precision, dependency, and efficiency. The usefulness, U, evaluates the correctness and validity of the customer operations.

$$U = 1 - \frac{\sum_{i=1}^{n} e_i}{n} \tag{2}$$

where e_i represents the ratios of error or nullify operations, $e_i \in [0, 1]$ and n is the operation number.

Precision, P, computes the average number of operations for completing the tasks.

$$P = \frac{\sum_{i=1}^{n} t_i}{n} \tag{3}$$

where t_i represents the number of operations to achieve the ith task and n is the task number.

Dependency, D, represents the behavior correlation.

$$D = \frac{\sum_{i=1}^{n} B_i \Rightarrow B_{i+1}}{n} \tag{4}$$

where B_i and B_{i+1} represent the i^{th} and $(i+1)^{th}$ behaviors, $\Rightarrow$ represents the sequence relationships, and n is the behavior number.

Efficiency, E, measures the average time of the customer for completing the tasks.

$$E = \frac{\sum_{i=1}^{n} w_i}{n} \tag{5}$$

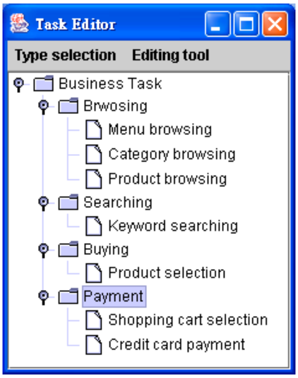
where w_i represents the time for completing the i^{th} task and n is the task number.

6 EC Application

The business process of business to consumer (B2C) classifies the customer interactions into four tasks, that is, browsing, searching, buying, and payment. The browsing task represents the behaviors that use hyperlink of the webpage to find the merchandises, such as, menu browsing or product browsing. The searching task represents the behaviors that use the search functions with the keywords to find the related products in the website. The buying task represents the behaviors that select the product into the shopping cart or send the order form to order the products. The payment task represents the behaviors that select the transporting vehicle and payment method. The customer then uses the above behaviors to buy merchandise from the EC system.

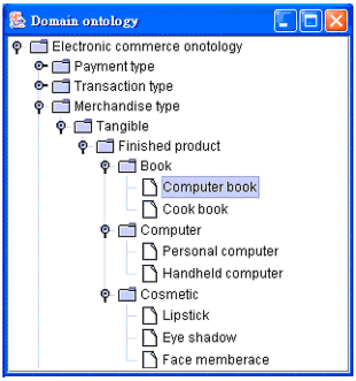
6.1 Task Editor

The intelligent monitoring system provides a graphical interface for the system administrator to edit the business tasks. Figure 5 lists the content of the business tasks. The editing frame displays the framework of domain ontology in the task editor. The merchandise of the EC ontology contains 3 classes, that is, book, computer, and cosmetic classes. Each class also contains the related subclasses with its class value. Fig. 6 shows the content of the domain ontology.



- Business Task
  - Brwosing
    - Menu browsing
    - Category browsing
    - Product browsing
  - Searching
    - Keyword searching
  - Buying
    - Product selection
  - Payment
    - Shopping cart selection
    - Credit card payment

Fig. 5. Business tasks



- Electronic commerce onotology
  - Payment type
  - Transaction type
  - Merchandise type
    - Tangible
      - Finished product
        - Book
          - Computer book
          - Cook book
        - Computer
          - Personal computer
          - Handheld computer
        - Cosmetic
          - Lipstick
          - Eye shadow
          - Face memberace

Fig. 6. Domain ontology

6.2 Action Supervisor

The action supervisor monitors the operations of the customer in the application. It collects the interaction operations form the interaction events and the HTTP message. It then excludes unnecessary operations and recognizes the preferred operation. Fig. 7 displays computation of the Bayesian belief network. The probability of $P(ST_2)$=P(Focus object)*P(Hyperlink clicking)=0.5*0.5=0.25. Moreover, the probability of $P(ST_1)$, $P(ST_3)$ and $P(ST_4)$ are zero. Finally, the action supervisor uses RBF neural networks to classify the behavior patterns according to the customer's operations.

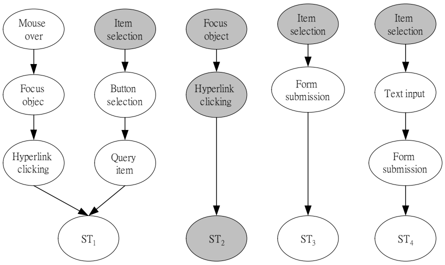
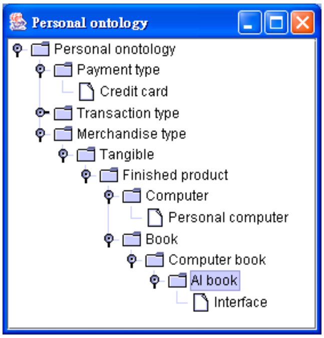


Fig. 7. Bayesian network computation

6.3 Behavior Analyzer

The behavior analyzer uses the task network to record the recognized behaviors also update the fuzzy behavior patterns. The behavior analyzer then constructs the personal ontology according to the products in the customer's behavior (Fig. 8).



- Personal onotology
  - Payment type
    - Credit card
  - Transaction type
  - Merchandise type
    - Tangible
      - Finished product
        - Computer
          - Personal computer
        - Book
          - Computer book
            - AI book
              - Interface

Fig. 8. Personal ontology

The behavior analyzer also evaluates the skill proficiency of the customer. Table 1 lists the tasks, behavior number, operation number, error operation, and visiting time of the customer interactions. The operations of the customer contain four tasks and the error ratio is 0.68, 0, 0, 0 and 0. Fig. 9 shows the computation of the skill proficiency.

Table 1. Customer operations

Task	Behavior number	Operations number	Error operations	Time
Browsing	3	10	2	42
Searching	1	3	0	8
Buying	2	4	0	12
Payment	1	2	0	9

$$U = 1 - \sum_{i=1}^{n} e_i = 1 - \frac{(2+0+0+0)}{19} = 0.89$$

$$P = \frac{\sum_{i=1}^{n} t_i}{n} = \frac{(10+3+2+4)}{4} = 2.75$$

$$D = \frac{\sum_{i=1}^{n} I_i \cap I_{i+1}}{n} = \frac{(0.5+0.5+0.5+0.3+0.7+0.1)}{6} = 0.43$$

$$E = \frac{\sum_{i=1}^{n} W_i}{n} = \frac{(42+8+12+9)}{4} = 17.75$$

Fig. 9. Skill proficiency evaluation

7 Conclusions

This work proposes an intelligent interface for customer behavior analysis in electronic commerce. The intelligent interface contains three modules, namely, the task editor, action supervisor, and behavior analyzer. The task editor provides an intelligent interface for the system administrator to define the business tasks and domain ontology. An action supervisor is defined as the information collector for monitoring the customer operations, excluding unnecessary operations, and recognizing the behavior patterns. The action supervisor uses the interaction message to extract customer operations, the Bayesian belief network to filter out redundant and irrelevant operations, and the RBF neural networks to recognize the behavior patterns. Finally, the behavior analyzer generates the customer behavior analysis information by measuring the behavior patterns, constructing the personalized domain ontology, and evaluating skill proficiency of the customer.

In summary, the proposed customer behavior analysis system exhibits the following features. First, the proposed system has an intelligent interface that allows system administrator to define the business tasks in human-computer interactions. The business tasks identify the relationships among the customer interaction, transaction behaviors, and business tasks definitely. The business tasks also can reduce the measurement complexity of customer interaction. The customer operation monitoring extracts the customer operations by analyzing the interaction protocol based on the customer interactions. The exclusion of unnecessary operations uses the Bayesian network to reduce the computation of irrelevant operations. The behavior pattern recognition uses the RBF neural networks to discriminate the behavior patterns based on customer operations. Moreover, the behavior model uses the task network and fuzzy behavior proximity to represent the behavior patterns and their relationships. Finally, the knowledge and skill proficiency evaluation uses EBL and proficiency metrics to discriminate end customer domain literacy and skill proficiency.

The work applied the proposed intelligent interface for customer behavior analysis to the EC. The application of the system demonstrates that the system can support system administrators in defining business tasks, and in turns can collect robust interaction information from customers. The analyzed customer behavior information can support the EC platform developer to refine the transaction workflow intelligently. The system possesses not only a customer behavior analysis but also an intelligent knowledge management in human-machine interface.

References

1. Aberg, J., Shahmehri, N.: The Role of Human Web Assistants in E-Commerce: An Analysis and a Usability Study. Electronic Networking Applications and Policy, 10, 2 (2000), 114-125
2. Brusilovsky, P., Cooper, D. W.: Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. Proceedings of International Conference on Intelligent User Interface, San Francisco (2002) 13-16
3. Deng. C. W.: The Design and Implementation of an Intelligent User Modeling Management System. Master Thesis, Fu-Jen Catholic University. Taiwan (2003)
4. Etgen, M., Cantor, J.: What Does Getting WET (Web Event-logging Tool) Mean for Web Usability. Proceedings of the 5th Conference on Human Factors and the Web. Maryland. Available at: http://zing.ncsl.nist.gov/hfweb /proceedings/etgen-cantor/
5. Fischer, G.: User Modeling in Human-Computer Interaction. User Modeling and User-Adapted Interaction, 11, 1-2 (2001) 65-86
6. Goecks, J., Shavlik, J. W.: Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. Proceedings of the International Conference on Intelligent User Interfaces (IUI-2000), New Orleans, (2000) 129-132
7. Kenichi, Y.: User Command Prediction by Graph-Based Induction. Proceedings of the Sixth International Conference on Tools with Artificial Intelligence, LA, CA, (1994) 732-735
8. Liton F., Schaefer, H.P.: Recommender System for Learning: Building User and Expert Models through Log-Term Observation of Application Use. User Modeling and User-Adapted Interaction, 10, 2-3 (2000) 181-207
9. Paganelli, L., Paterno, F.: Intelligent Analysis of User Interactions with Web Applications. Proceedings of the International Conference on Intelligent User Interface, San Francisco, (2002) 111-118
10. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn. Prentice Hall, New Jersey, (2003)
11. Seo Y., Zhang B.: A Reinforcement Learning Agent for Personalized Information Filtering. Proceedings of the International Conference on Intelligent User Interface (IUI-2000), New Louisiana, (2000) 248-251
12. Shaw, M.J., Blanning, R., Strader, T., Whinston, A.: Handbook on Electronic Commerce. Springer-Verlag, Berlin Heidelberg New York (1999)

Integration of an Interactive Multimedia Datacasting System

Wei Li, Hong Liu, and Gilles Gagnon

Communications Research Centre, 3701 Carling Ave., Ottawa, ON, CANADA K2H 8S2
{wei.li, hong.liu, gilles.gagnon}@crc.ca

Abstract. Hardware and software integration of a framework for interactive multimedia datacasting applications is presented in this paper. The hardware components of this framework consist of a data server, an IP encapsulator (IPE), a multiplexer, data receivers and a return channel. Various integration software modules were implemented to support the hardware configuration, and these include IPE flow control, encoding/decoding rate control, opportunistic data injection etc. Server/client issues were investigated to support various applications such as unicast/multicast services, web accessing and media delivery rate control over digital television (DTV) broadcast infrastructure. The main contributions of this paper are: 1) the design of an interactive end-to-end datacasting framework; 2) the integration of hardware and software modules for this system; and 3) Unicast and Multicast data transmission modes. This framework was tested in a terrestrial broadcasting environment with various multimedia datacasting applications.

1 Introduction

In North America, the Advanced Television Systems Committee (ATSC) Digital Television (DTV) standard [1] has been adopted for over-the-air delivery of digital television programs to homes. The ATSC system is capable of carrying multiple standard definition television (SDTV) programs or a single high definition television (HDTV) program with the associated sound, as well as ancillary data information in a single 6 MHz terrestrial broadcasting channel. Encapsulation of IP data into an ATSC transport stream (TS) allows the transmission of IP-based multimedia and data information over DTV channels [2]. Recent implementations of interactive multimedia delivery systems for specific applications can be found in [3], [4] and [12]. In [12], a multimedia broadcasting system using a DTV channel is deployed by using a scheme based on the Digital Storage Media-Command and Control (DSM-CC) functionality and a new protocol which efficiently conveys multimedia information having the same characteristics as HTML files. The real-time multimedia broadcasting scheme proposed in [3] enables distortion free TV-reception in fast moving vehicles with the adoption of MPEG-4 based digital audio broadcasting/digital multimedia broadcasting (DAB/DMB). A streaming system for interactive television broadcast using MPEG-4 and MPEG-7 in edition and transmission of multimedia programs is proposed in [4]. This system is capable of providing two transport streams per program, one via a program multiplexer and the

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 325-334, 2004.

other via a data multiplexer. These two multiplexers together deliver all the types of data mentioned in the ATSC data broadcast specification.

A seamless integration of hardware and software modules is essential for an interactive multimedia delivery system to be effective. Well-supported software modules can lead to more efficient resource utilization towards hardware components. The selection of hardware components determines the system configuration and software applicability. In this paper, we propose the architecture for an interactive multimedia datacasting system. Various software modules were implemented to support this hardware configuration. The goal of the project is to investigate, develop and demonstrate means of utilizing the DTV infrastructure to provide interactive broadband multimedia services. Such a system can be used by broadcasters to provide a variety of program related or non-program related services such as Internet access, file transfer, audio/video multicasting etc. In addition, such a system could provide remote and rural communities with interactive broadband access to a number of services such as education, medical consultation, e-learning, web access, banking and many others. Dedicated DTV channels could be used to provide broadband access in rural and remote areas, as there is ample available spectrum in these regions.

This paper is organized as follows: Section 2 introduces the system architecture that is being developed. This system is composed of the following components: data server, IP Encapsulator, transport stream multiplexer, data receiver and return channel. Issues related to the development and integration of software required to support the hardware framework are discussed in section 3. A number of multimedia datacasting applications related to this framework are described in section 4. Conclusions and future considerations are discussed in section 5.

2 System Architecture

The interactive multimedia datacasting system consists of a broadcast (downlink) subsystem and a return channel (uplink) subsystem. This system is illustrated in figure 1. According to user requests or the broadcaster's predefined scheduling, the downlink subsystem serves to combine various data and multimedia information onto a MPEG-2 transport stream (TS) [5] which is then modulated to one of the television channels. The IP Encapsulator (IPE) encapsulates the IP data coming from the content server onto an MPEG-2 TS according to the Multi-Protocol Encapsulation (MPE) or the data piping method of the ATSC Data Broadcast Standard A/90 [2]. Once the IP data is converted to MPEG-2 TS, it is sent to a transport stream multiplexer (TSM) where other sources of MPEG-2 transport streams can be added. The resulting MPEG-2 TS can carry contents such as other encapsulated data or compressed MPEG-2 video/audio programs meant for broadcasting. Many services and programs can be multiplexed together as long as the 19.4 Mbps limit imposed by the system is respected. The signal from the multiplexer is then fed to the 8VSB modulator/transmitter. The receiver or set top box (receiver/STB) at the user's premise demodulates the received RF signal, recovers the MPEG-2 TS, extracts the media types and dispatches the demultiplexed media packets to the client application accordingly. This subsystem has the ability to deliver Quality of Service (QoS) transmission by setting a maximum as well as a guaranteed data rate at the IPE for an individual transport stream according to its program identification (PID).

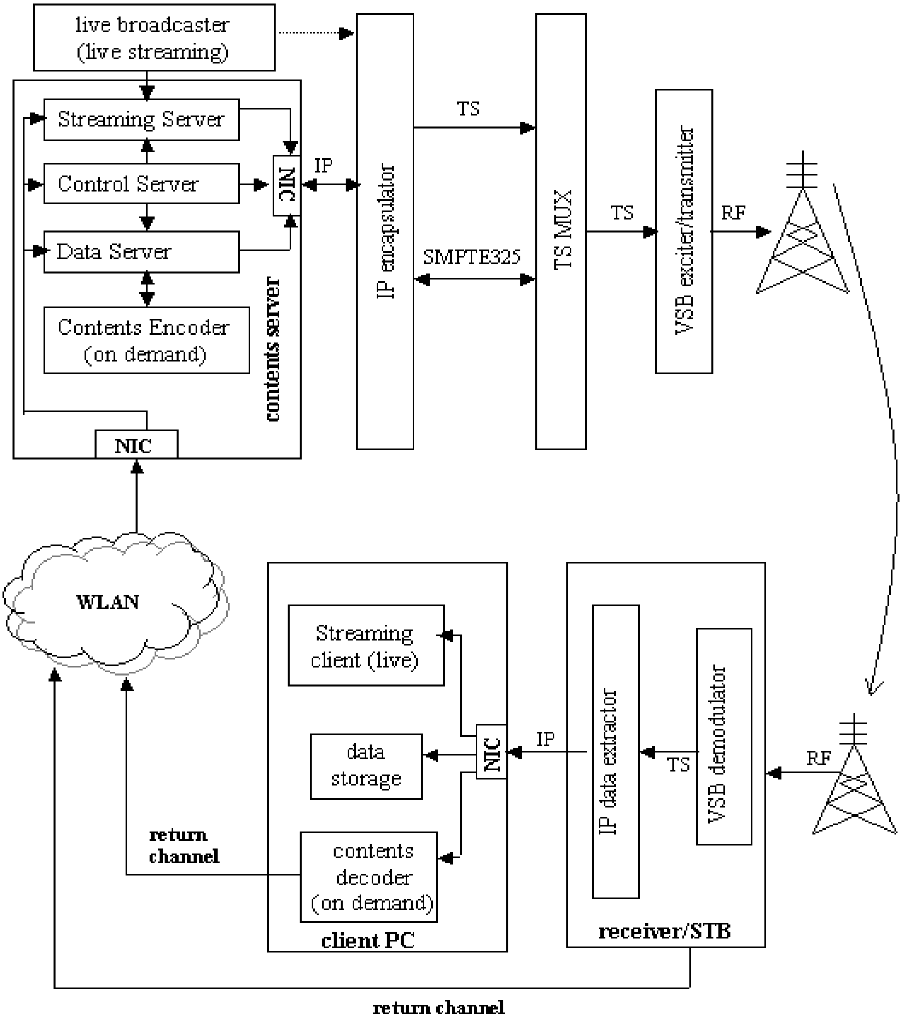


Fig. 1. Diagram of the interactive datacasting system

For laboratory test purposes, the uplink or return channel subsystem uses a wireless LAN (WLAN) in a unidirectional mode to send interactive requests to the server. With directional antennas, this return channel should be capable of offering the range and throughput required for field tests. The service requests (such as Web access, audio/video streaming etc.) that are to be processed by the server, the broadcasting system status and statistics such as the real-time control protocol (RTCP) reports are sent by client PC to the server via the return channel.

Investigations on how to provide a return channel using the television UHF band are ongoing since the current North American DTV standard does not include a return channel to support interactive applications. In Europe, the DVB family includes a wireless terrestrial return channel called DVB-RCT (Return Channel Terrestrial). We are considering adapting the DVB-RCT system for use with the North American ATSC A/53 standard [1].

The interactive datacasting system has been developed in a PC environment under the Linux and Windows operating systems. The remainder of this section describes the main functional modules individually.

2.1 The Content Server

High capacity and proficiency in providing interactive video, audio and data services were taken into consideration in the design of the content server. This server is the integrated result of the following components:

- **Content encoder** is a hardware encoder which converts raw video/audio data to compressed data and stores it in the server's hard disks. The transmission of the encoded data to a client is carried out via the data server in response to the client's demand.
- **Streaming server** is a software reflector that allows the media streaming (e.g. MPEG-4 audio/video, MP3 audio) from a live broadcaster source to real time streaming protocol (RTSP) clients. It also serves as an administrator to add/delete/monitor the connection status, to receive the feedback requests from client machines and to react to those requests. The output of the streaming server is IP data and is sent to the IPE.
- **Data server** gets miscellaneous data such as audio, video, text, etc. from outside networks and feeds them to the contents encoder when there is a demand.
- **Control server** plays an important role in controlling the operations of the data server and streaming server. It receives the requests and feedback information from clients through the return channel and manages the mentioned servers' operations so as to guarantee high quality of service.

2.2 Live Broadcaster

The live broadcaster is a software module for encoding and streaming media (e.g. MPEG-4 audio/video) over the network. To broadcast live content, we can either use the direct media to unicast/multicast to clients, or we can use the streaming server to reflect/manage the live broadcast contents to clients. The latter is achieved via the insertion of the media and network configuration information such as a Session Description Protocol file (SDP [6]) into the content directory of the streaming server.

2.3 IP Encapsulator (IPE)

The IPE is a hardware device which provides an efficient method for inserting or multiplexing numerous IP data services, delivered over Ethernet, into an ATSC compliant transport stream (e.g. MPEG-2 TS) for data broadcasting. The IPE features a 100 Mbps Ethernet input and an ASI interface for the connection to a multiplexer or a modulator. The built-in application software offers an intuitive GUI that provides configuration and monitoring of outgoing transport data streams. Parameters such as QoS and unicast or multicast mode of operation can be specified for each individual PID.

2.4 Transport Stream Multiplexer (TSM)

The transport stream multiplexer combines a number of MPEG-2 transport streams into a single stream at 19.39 Mb/s. The multiplexer supports opportunistic data flow control by means of SMPTE 325M [10] to maximize the bandwidth usage up to 100 percent capacity by replacing NULL TS packets with IP data.

2.5 8-VSB Exciter / Transmitter

A modulator is composed of an 8-VSB exciter and a transmitter. For terrestrial broadcasting, the multiplexed MPEG-2 TS is sent to the 8-VSB exciter [1] using the SMPTE 310M [11] synchronous serial interface standard. Reed-Solomon coding and 8-VSB modulation are performed in the exciter.

The DTV transmitter is connected to the output of the 8-VSB exciter; the transmitter modulates the incoming signal into a 6-MHz RF band and sends out the signal to the antenna. For field trials we have access to a 30 KW ERP DTV transmitter situated approximately 25 Km south of Ottawa.

2.6 Data Receiver / STB

The data receiver provides the following functions: RF tuning, demodulation of the received signals to an MPEG-2 TS, extraction of the IP data from the TS and forwarding of the data to the client PC and specific application. These functions could be included into a set-top box (STB) connected to a PC through Ethernet.

The actual receiver consists of two functional modules:

- **VSB demodulator** transforms the received RF signal to an MPEG-2 transport stream.
- **IP data extractor** extracts IP data from the MPEG-2 TS and relays the IP data packets to their destinations.

2.7 Client PC

The interactive client PC provides end-users with services such as Internet access, media downloading, Video on Demand (VoD) and live streaming playback as well as service requests to the data server. Its functional modules are listed below:

- **Streaming client** is a software decoder/player for receiving encoded live media streaming (e.g. MPEG-4 live audio/video).
- **Data storage** provides support for media on demand downloading services at the client end. There are a number of different types of media to be stored at the client PC. Examples of media types include discrete files, image bitmaps, Web pages, Java applets etc.
- **Content decoder** provides on demand audio/video decoding and playback. We use software-based decoders and apply a simplified probe-based rate control approach [7] to adjust the transmission rate from the server in accordance with network availability .

2.8 Return Channel

Interactivity is provided through a wireless return channel that could be in the UHF band. In Europe, DVB-RCT [8] has been defined for a terrestrial return channel, while the ATSC DTV Standard does not include such a return channel component yet.

It is to be noted that different types of return channels can be implemented and used simultaneously with others. For cost efficiency, low throughput and, if we assume that most calls are local, 56Kbps modems can be utilized in conjunction with a public switched telephone network.

To reduce cost and complexity even further, the system supports a number of multicast streaming applications that do not require a return channel. The only requirement is to let the client applications know the multicast address of the streaming service they want to access. Announcement information similar to program guides can be implemented using the Session Announcement Protocol (SAP) and can be broadcasted on a regular basis to inform clients of which multicast services are available that do not require a return channel.

3 Software Implementation Considerations for the System

3.1 IPE Flow Control

The IPE flow control is a software module in the IPE providing real-time IPE buffer utilization monitoring to the streaming server. It informs the streaming server of the IPE buffer fullness by sending back IP multicast packets at a user-defined interval. The streaming server uses the information carried in the packets to adjust its outgoing data rate to ensure bandwidth efficiency.

Figure 2 illustrates the control packet structure. The IPE buffer status information is sent out from the IPE to the server in a UDP multicast packet. The destination multicast IP address, port number and transmission interval are configurable through the control module GUI.

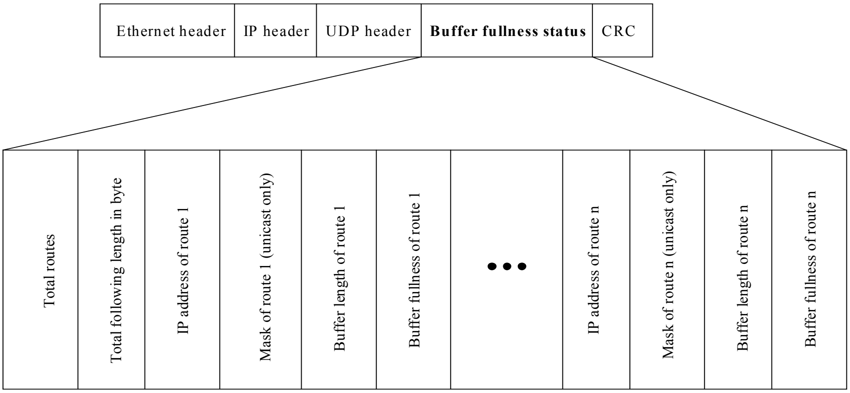


Fig. 2. The flow control multicast packet structure

3.2 Probe Based Content Encoder / Decoder Rate Control

The content encoder produces MPEG-2 TS data from raw audio/video data and transfers the TS data to the contents decoder of the client PC via a DTV channel. The contents encoder/decoder pair plus the corresponding data conversion, modulation, transmission techniques compose the media on demand service of our system. To ensure the quality of service during media transmission, we implemented a probe-based rate control approach for the feedback control at the content encoder and decoder.

The probe-based approach has been refined based on experimental results. Specifically, the source probes for the available network bandwidth by adjusting the sending rate in a way that maintains the packet loss ratio below a certain threshold.

Consistent with the RTP/RTCP standard [9], the contents decoder periodically sends RTCP packets to the encoder via the simulated return channel. The RTCP packet contains the packet loss ratio P_{loss} obtained during a certain time interval. Considering the hardware MPEG-2 encoder/decoder chipset used in our system, the proposed rate control protocol performs the following interactions between the encoder and decoder:

Contents encoder

- The encoder starts at its preset rate to encode data to the sending buffer. A packet sequence number is incremented and put into each RTP data packet.

- Upon the receipt of the RTCP feedback packet containing the packet loss ratio P_{loss} from the decoder, the encoder will adjust its waiting time interval between two successive encoded data packets before sending them out from the sending buffer. The time interval of the two adjacent packets Δt is calculated according to the following rule:

```
if ( $P_{loss} \le P_{threshold}$ )
    $\Delta t = \max\{t - DecR, 0\}$ ;
else
    $\Delta t = \max\{t \times IncR, \Delta T\}$ .
```

(where *DecR*, *IncR*, and ΔT represent the additive decrease factor, multiplicative increase factor and the minimum time interval respectively)

Contents decoder

- The received packets are monitored by the extraction of their sequence numbers.
- After a certain time interval, the decoder sends back a RTCP packet to the encoder with the packet loss ratio observed during this time period.

3.3 Opportunistic Data Injection

The ATSC standard mandates a fixed data transmission rate of 19.39 Mbps per channel. While MPEG-2 encoders usually operate at constant bit rate (CBR) mode, there can be spare capacity to insert data up to the maximum transmission rate. To accommodate the need for a full bandwidth usage, the SMPTE 325M standard [10] is adopted in the TS multiplexer and the IP Encapsulator for opportunistic data injection.

The opportunistic data injection module removes NULL packets and inserts IP data packets from the IPE to the TSM in order to utilize the available bandwidth. The TSM provides buffer storage for the data packets. When the data is sent out of the TSM, space becomes available in the buffer. Requests will then be sent from the TSM to the IPE in order to get additional packets to fill its buffer.

3.4 Raw Socket Utilization

The raw socket programming technique is utilized in the data receiver/STB module to relay incoming encapsulated IP data to the destined client. The raw socket provides the capability of avoiding the IP datagrams protocol field being processed by the kernel, thus reducing the overhead caused by the deletion/addition operations normally required on the received packets at the data receiver.

Upon receiving the TS data from the VSB demodulator, the IP data filter extracts the IP data from the TS then sends the data directly to the destination client on the Ethernet by using the raw socket. In this way, it is not required to get rid of the IP headers of the received packets and re-packetize them before forwarding them to their destination.

4 Applications

We have successfully integrated hardware components and software modules into an interactive multimedia datacasting framework. Various applications are carried out with this system under both Linux and Windows operating systems. Lab tests have proved the feasibility of this framework and the applicability in various datacasting applications.

The following applications are currently available:

- MPEG unicast and multicast transmission;
- MPEG-1 (VCD) and MPEG-4 streaming;
- Internet access (Web browsing, ftp downloading etc.);
- Media on demand (audio, video and text files);
- IP Encapsulator (IPE) flow control;
- Opportunistic data insertion between the TS multiplexer and the IPE.

5 Conclusion and Future Work

An interactive end-to-end datacasting system using the ATSC DTV infrastructure was designed in the context of offering broadband access to rural and remote communities. Hardware and software intergration at both the server and client ends was performed to allow easy access to multimedia services. Laboratory tests showed the feasibility and sound performance of this system. Required future work includes: 1) adaptive encoding rate control for MPEG video, 2) channel QoS study, 3) return channel protocol study, 4) Data caching suitable for audio/video streaming, 5) data carousel implementation etc.

References

1. ATSC Standard A/53, "ATSC Digital Television Standard," September 1995.
2. ATSC Standard A/90, "ATSC Data Broadcast Standard," July 2000.
3. M. Grube, P. Siepen, C. Mittendorf, M. Boltz, and M. Srinivasan, "Application of MPEG-4: Digital Multimedia Broadcasting," *IEEE Trans. on Consumer Electronics*, **Vol. 47**, No. 3, pp. 474-484, August 2001.
4. J. Yang, Y. Kim, S. Ahn, M. Park, C. Ahn, J. Seok, Y. Lim, and K. W. Lee, "A Design of a Streaming System for Interactive Television Broadcast," *IEEE International Symposium on Circuit and Systems*, Geneva, pp. III 559-562, May 28-31, 2000.
5. ISO 13818-1, "Information Technology –Generic Coding of Moving Pictures and Associated Audio: Systems," Nov. 13, 1994.
6. RFC 2327, "Session Description Protocol," April 1998.
7. D. Wu, Y. T. Hou, W. Zhu, H. J. Lee, T. Chiang, Y. Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuit Syst. Video Technology*, **Vol. 10**, pp. 923-941, September, 2000.
8. ETSI EN 301 958, "DVB: Interaction Channel for Digital Terrestrial Television (RCT) Incorporating Multiple Access OFDM," March 2001.

9. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-time Applications,” Internet Engineering Task Force, RFC 1889, Jan. 1996.
10. SMPTE 325M, “Digital Television – Opportunistic Data Broadcast Flow Control”, 1999.
11. SMPTE 310M, “Synchronous Serial Interface for MPEG-2 Digital Transport Stream”, 1998.
12. L. Atzori, F. Natale, M. Gregorio, and D. Giusto, “Multimedia Information Broadcasting Using Digital TV Channels,” *IEEE Trans. on Broadcasting*, **Vol. 43**, No. 3, pp. 242-251, September 1997.

The Exploration and Application of Knowledge Structures in the Development of Expert System: A Case Study on a Motorcycle System

Kuo-Wei Su[1], Sheue-Ling Hwang[2], and Yu-Fa Zhou[2]

[1]Assistant Professor, Department of Logistics Management,
Takming College, Taipei, Taiwan, R.O.C
[2]Professor, Department of Industrial Engineering and Engineering Management,
National Tsing-Hua University, Hsinchu, Taiwan, R.O.C.
kwsu@mail.takming.edu.tw

Abstract. This study proposes an expert system that applied in Taiwan motorcycle's fault detection and maintenance. An expert's knowledge structure was explored, to construct a knowledge base which has the expertise and then to build a friendly human-computer interface. This study will examine the knowledge structures of a novice group and an expert group by using knowledge network organizing method. And the similarity between theses two groups is assessed from network structure indices. The expected results will indicate the knowledge structure of an expert group is more organized than a novice group and then a suitable expert system will be developed. To develop the expert system for the fault detection and maintenance can not only promote the maintenance efficiency and quality, but also make up for the shortage of Taiwan motorcycle's maintainers. It also, of course, can serve as a training tool for maintainers and shorten learning time effectively.

1 Introduction

According to the statistics of the MOTC (ministry of transportation and communications) in 2002, there were 12,245,541 motorcycles in Taiwan. Though Taiwan motorcycle's density is the highest in the world, the motorcycle maintenance quality is not really the same level. It's predictable that the high maintenance level will be required in local market after opening the import of heavy and electrical motorcycle.

For the reason of maintenance skill enhancement, the study will adopt some techniques to develop the expert system for assisting the maintenance tasks proved in the previous study [17]. The failure modes and effects analysis (FMEA) technique can be used in the research for finding out a wealth of useful information that can be assisted in achieving the expected system reliability [17]. To use the FMEA tool, the critical maintenance points will be found and worthily established into the expert system (ES). For constructing ES's core - knowledge base, pathfinder network is the first procedure for exploring the organization of expert's and novice's domain knowledge. Pathfinder relies upon a mathematical algorithm for creating a graphic representation based on implicit proximity data. This process is examined by using

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 335-340, 2004.

KNOT (Knowledge Network Organization Tool) [18]. The similarities between novice's and expert's networks are estimated from closeness (PFC or C), proximities (PRX), and graph-theoretic distance (GTD) indices. It is expected that the knowledge structure of an expert group is more organized than a novice group. An expert system will be provided to novice to facilitate the maintenance tasks and make their knowledge structure closer to expert's knowledge structure.

In this paper, an ES's prototype is addressed to simulate human expertise during the problem solving by incorporating artificial intelligence and coupling various descriptive knowledge, procedural knowledge and reasoning knowledge involved in the industry knowledge. The system is developed through employing Visual Rule Studio, a hybrid expert system shell, as an ActiveX Designer under Microsoft Visual Basic 6.0 environment since it combines the advantages of both production rules and object-oriented programming technology [9]. Both forward chaining and backward chaining are used collectively in the inference process. The constructing process of ES is also served as a guide for other similar domain-specific knowledge management system.

2 Relevant Analysis

2.1 Failure Modes and Effects Analysis (FMEA)

FMEA originated as a formal methodology in the 1960s when demands for improved safety and reliability extended studies of component failures to include the effects of the failures on the systems of which they were a part. By the FMEA report, the critical effects and risk priority will be easily found and provide potential failure mode, failure effect, and failure cause and to prevent system breakdowns [17].

2.2 Pathfinder Network Analysis

After obtaining key failure mode, the corresponding test questions can be developed as input of KNOT. Then the novice's and expert's knowledge structure and concept diagrams can be obtained, so as to acquire expert's knowledge to improve novice's cognitive representation.

According to Goldsmith et al., (1991) a better methodology for assessing the expert's and novice's knowledge domain is a structural assessment approach. Three distinct steps comprise the structural approach: (a) knowledge elicitation, (b) knowledge representation, and (c) evaluation of a learner's knowledge representation [11] [16].

2.3 The Basic Components of an Expert System

A term of the ES usually made up of domain expert, programming designer, and knowledge engineer and the knowledge engineering is the central role to build up the ES. Knowledge engineer utilized the method of knowledge structure measurement to acquire strategic decision and knowledge into the knowledge base and continuously tested and revised to develop more effective ES [10] [12].

A complete ES consists of three basic components. The first component is the knowledge base, which has been developed using a commercially available expert system shell called VRS (Visual Rule Studio) in the study. It is a hybrid application development tool that installs an integral part of Microsoft Visual Basic 6.0 as an Active Designer [9]. The second is the inference engine, which drives the knowledge base through reasoning processes which are similar to experts. VRS supports three types of inference strategies: Backward-Chaining, Forward-Chaining and Hybrid-Chaining.

The last is the user interface, which is the means by which the user communicates with the knowledge base (or virtual expert). It also allows user to question the ES and to provide some advice. A good user interface is a necessity for the success of knowledge-based expert systems. The relationships between the user, computer system, user model and system model, via the user interface, and how the design of a good user interface should take into account both these two models [1]. Usability is a well-known and well-defined concept in the HCI research. It has been defined as ease of learning, efficiency of use, memorability, error rates, and preference in the HCI area. The three-component framework including user, product, activity, and environment has long been accepted as the principal components in a human-machine system upon which good system design depends [8] [15].

3 Method

3.1 Participants Definition

In this research, three participants are sampled in each expert group and novice group. The novice's participants were asked to read a paper about the introduction of the key motorcycle component. Then the novices are asked to do a test to obtain their knowledge structure. The sample of the expert group and novice group must have the unity of a sample that is described as follows:

Expert: the participants who have over five years of working-experience and professional license are sampled. They are working in different motorcycle shops.

Novice: the participants who have no working-experience are sampled. They are graduate students of Industrial Engineering department at National Tsing-Hua University in Hsinchu city.

3.2 Experimental Materials

The examination questions of motorcycle technique test can be used to evaluate and compare the difference of knowledge structure between experts and novices. This research will analyze three distributors which have about 10% shares in Jilong city.

The component related data of every month orders is collected from 2002.06.01 to 2003.02.01.

The computer program of KNOT is employed for pathfinder network analysis and to draw the knowledge structure of novices and experts with different experienced level. The expert system is developed through employing VRS, a hybrid expert system shell, as an ActiveX Designer under Microsoft Visual Basic 6.0 environment since it combines the advantages of both production rules and object-oriented programming technology.

3.3 Procedure

The framework construction procedure is described below and the flow chart of research is shown in Figure 1.

Stage1. To find out the key failure model:

As shown in Table1, engine and electric mechanism have significant higher component order cost than others. Consequently, electric and engine mechanism will be analyzed in FMEA. Finally, to find out PRN (Risk Priority Number) for a component failure mode is developed from the occurrence probability of the failure mode, the severity of its failure effect, and the probability of the failure being detected.

Table 1. The component cost of order

Motorcycle framework	Engine	Electrical	Body	Exhaust
Cost (NT dollars)	33,061	44,719	29,857	7,209

Stage2. Pathfinder network analysis:

The KNOT provides the RATE program allow measure is based on a kind of transitivity assumption, i.e., if two concepts have similar relationships with other concepts, then the two concepts should be similar to one another. Then KNOT through data transformations and processing procedures to draw the knowledge structure of such novices and experts and calculate the PRX, GTD and PFC indices. Then the difference of knowledge structure chart between the novices and experts can be compared.

Stage3. To build a prototype of ES:

The VRS is an expert system shell and a hybrid application development tool that installs an integral part of Microsoft Visual Basic 6.0 as an Active Designer. A knowledge engineer uses the VRS to build the ES knowledge-base and to design ES interface.

Stage4. Evaluation for the system:

After constructing the ES, it is provided for the novice to use and get their knowledge structure in a while. Furthermore, pathfinder network analysis is reused to compare the similarity between experts and novices and to verify the feasibility and effects of the ES.

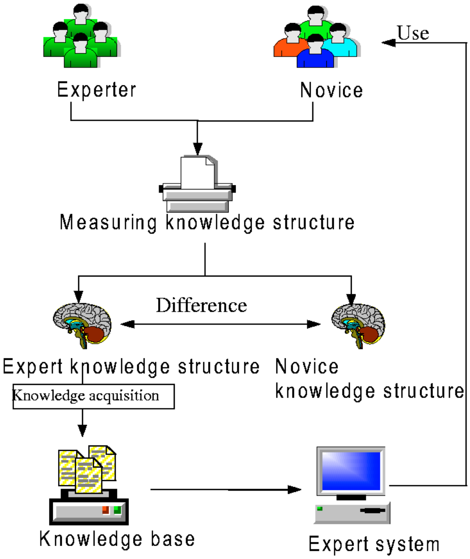

Use
Experter
Novice
Measuring knowledge structure
Difference
Expert knowledge structure
Novice knowledge structure
Knowledge acquisition
Knowledge base
Expert system

Fig. 1. The flow chart of the research

4 Work in Progress

This study will develop the ES and explore its effects, which utilizes experts' knowledge from their knowledge structure. It also can serve as a training tool for maintenance workers and shorten training time effectively. The above process will be studied in the future and there are a few anticipating results in the following:

- To utilize the KNOT to draw the knowledge structure of such novices and experts and calculate the PRX, GTD and PFC indices. Then to prove the experts' knowledge structure is better than the novices before using the expert system.
- Some diagrams for the repair illustration are added to make maintainers' understand and complete diagnosis and repair of the breakdown rapidly.
- To utilize expert knowledge structure in designing the expert system that applied to motorcycle's fault detection and maintenance. It will help for novice to promote maintenance technique and the capability of fault detection.
- Finally, to verify the feasibility and effects of the ES after novice uses the expert system.

However, this suggesting possible research directions and expectation may necessarily hold true and there is no alternative but to wait for future research and further development.

References

1. Berrais, A. (1997). Knowledge-based expert system: user interface implications. Advances in engineering Software, 28, 31-41.
2. Frederick Hayes-Roth (1992). Knowledge Systems: An Introduction. Library Hi Tech,, 10, no. 1-2.
3. Goldsmith. T.E., Johnson, P.J., & Acton, W.H.(1991). Assessing structural knowledge. Journal of Educational Psychology, 83, (1), 88-96.
4. Gonzalvo, P., Canas, J.J., & Bajo, M. (1994). Structural representations in knowledge acquisition. Journal of Educational Psychology, 84, (4), 601-616.
5. Hsiao-Fang Lin (2001), A Study of Knowledge Representation and Concept Learning: Using Pathfinder Network as Assessment Tool, Journal of Education and Psychology, 24, 229-262.
6. Hsiao-Fang Lin & Min-Ning Yu (2001), An Evaluative Research Study on Learning the Mathematics Concept of Algebra by High School Students–Using the Quadratic Equation as Example, Journal of Education and Psychology, 24, 303-326.
7. J.L. Gordon (2000). Creating knowledge maps by exploiting dependent relationships. Knowledge-Based Systems, 13, 71-79.
8. Jiyoung Kwahk, Sung H. Han*(2002), A methodology for evaluating the usability of audiovisual consumer electronic products, Applied Ergonomics, 33, 419–431.
9. K. W.Chau*, C. Chuntian, C.W. Li (2002), Knowledge management system on flow and water quality modeling, Expert Systems with Applications, 22, 321-330.
10. Lin Yi-Chun (2002), Apply Expert System to the Fault Detection of HVAC, A master paper in National Taiwan University of Science and Technology Department of Electrical Engineering
11. Min-Ning Yu, Hsiao-Fang Lin & Chia-Yen Tsai (2001), Cognitively Diagnostic Assessment of Mathematical Knowledge Structures of Elementary School Children, Journal of Education and Psychology, 24, 263-302.
12. P. J. Geraghty (1993), Environmental Assessment and the Application of Expert Systems: an Overview, Journal of Environmental Management, 39, 27-38.
13. R. Bromme and H.Tillema (1995). Fusing experience and theory: The structure of professional knowledge. Learning and Instruction, Vol.5, pp.261-267.
14. Statistics of Transportation & Communications (2002), http://www.motc.gov.tw.
15. Sung H. Han*, Myung Hwan Yun, Kwang-Jae Kim, Jiyoung Kwahk (2000), Evaluation of product usability: development and validation of usability dimensions and design elements based on empirical models, International Journal of Industrial Ergonomics, 26, 477-488.
16. Steven J. McGriff, & Dr.Peggy Van Meter (2001). Measuring Cognitive Structure: An Overview of Pathfinder Networks and Semantic Networks. http:// www. personal.psu.edu/faculty/s/j/sjm256/portfolio/kbase/Theories&Models/Cognitivism/Cognitive-Structure.pdf
17. Su, k. W., Hwang, S. L., T. H.(2000a). Reliability-centered maintenance management: a case study on a public bus system. International Journal of Industrial Engineering-Theory, Application, and Practice, 7 (3), 219-228.
18. William H. Acton, Peder J. Johnson, and Timothy E. Goldsmith (1994). Structural Knowledge Assessment: Comparison of Referent Structures. Journal of Educational Psychology, 86, (2), 303-311.

Binary Decision Tree Using K-Means and Genetic Algorithm for Recognizing Defect Patterns of Cold Mill Strip

Kyoung Min Kim[1,4], Joong Jo Park[2], Myung Hyun Song[3], In Cheol Kim[1], and Ching Y. Suen[1]

[1] Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1455 de Maisonneuve Blvd. West, Suite GM606, Montreal, Canada H3G 1M8
{kkm, kiminc, suen}@cenparmi.concordia.ca
[2] Department of Control and Instrumentation Engineering, Gyeongsang National University, 900, Gazwa-dong, Chinju, Gyeongnam, 660-701, Korea
[3] Department of Electric Control Engineering, Sunchon National University, 315, Maegok-dong, Suncheon, Chonnam, 540-742, Korea
[4] Department of Electrical Engineering, Yosu National University, San 96-1, Dunduk-dong, Yeosu, Chonnam, 550-749, Korea

Abstract. This paper proposes a method to recognize the various defect patterns of a cold mill strip using a binary decision. In classifying complex patterns with high similarity like these defect patterns, the selection of an optimal feature set and an appropriate recognizer is a pre-requisite to a high recognition rate. In this paper GA and K-means algorithm were used to select a subset of the suitable features at each node in the binary decision tree. The feature subset with maximum fitness is chosen and the patterns are divided into two classes using a linear decision function. This process is repeated at each node until all the patterns are classified into individual classes. In this way, the classifier using the binary decision tree can be constructed automatically, and the final recognizer is implemented by a neural network trained by standard patterns at each node. Experimental results are given to demonstrate the usefulness of the proposed scheme.

1 Introduction

To produce a cold mill strip of high quality, it is important to extract the defects on the surface of cold mill strip rapidly in the manufacturing process. So, efficient methods for the recognition and extraction of defect patterns of cold mill strips have been studied[1]. Recently, a pattern recognition method to substitute a defect extraction system using a one dimensional reflected laser signal.

The conventional method to recognize the defect patterns is to extract good features experimentally after preprocessing the image acquired from a CCD camera and then recognizes the patterns in a single step by inputting all the features to a neural network. But this method has two problems when the characteristics of the defect

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 341-350, 2004.

patterns are considered. Firstly, because the shapes of the defect patterns are complex and irregular, the recognition rate of defect patterns is sensitive to the kinds of selected features. And also despite the good separability of the features, they may interfere with each other when used together. So, the fitness of the selected feature subset cannot be guaranteed if the features are selected experimentally. Secondly, because there exist some similar classes of defect patterns, which can be classified into the same group, classifying all the patterns in only a single step results in a high classification error.

To overcome these problems, we propose a multi-stage classifier like a decision tree, which repeats decisions so as to classify patterns individually. The decision tree classifier makes fast and exact decisions by dividing the complex and global decisions into several simple and local decisions [5][8].

For an efficient and accurate classification, an optimal or near-optimal feature subset within the feature space needs to be selected at each decision node [2]. There are three potential advantages in applying a method of selecting a subset of features for an input to the classification. Firstly, the performance of the classifier can be improved by reducing the possible inputs to a set of relevant uncorrelated variables. Secondly, the selected smaller set of features reduces both the time complexity and the processing time needed to produce the feature set, thus speeding up the response time of the system. Finally, since there is a direct relationship between the dimensionality of a problem and the size of the example set needed to adequately cover the problem space, the reduction of the feature set indicates that a smaller off-line training set can be used with all the secure benefits in terms of data collection and training times.

In this paper, GA and K-means algorithm are used to find a subset that yields the lowest error rate of a classifier. This search method has the advantage of producing a nearly optimal solution quickly. The fitness function used to evaluate the selected subsets of features is the error estimator for the linear decision function, which is also produced by GA and K-means algorithm. They make a linear decision function, whose dimension is that of the selected feature space, and searches for a linear decision function, which minimizes the classification error with which the fitness of the selected feature subset is calculated. Finally, the feature subset with maximum fitness is chosen, and the patterns are classified into two classes by the linear decision function. This process is repeated at each node until all patterns are classified respectively into the individual classes. In this way, the binary decision tree classifier is constructed automatically. After constructing the binary decision tree, the final recognizer is accomplished by a neural network, which learns from a set of standard patterns at each node.

This paper introduces the binary decision tree and presents methods of both generating a linear decision function and selecting a feature subset using GA and K-means algorithm. Then, an automatic method of constructing the binary decision tree is described. And finally, the two classifiers are applied to recognize the defect patterns of a cold mill strip.

2 Construction of Binary Decision Tree Using GA and K-Means Algorithm

If only the necessary features are used at each node of binary decision tree classifier, both the accuracy and the reliability of the classifier are increased. So the problem is to select a valid feature subset from the entire feature set, the feature selection problem.

GA has a higher probability of finding the global optimized solution than other conventional optimization algorithms because it searches for the multiple global solutions simultaneously. Thus the optimal feature subset can be selected effectively[1][3][4].

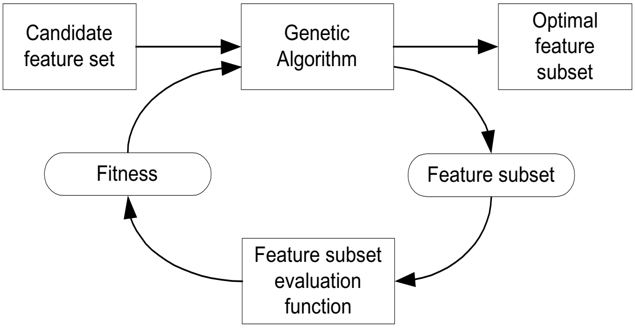


Fig. 1. Processing block diagram of feature selection

Fig. 1 shows the process of selecting the optimal feature subset by GA. In GA, the chromosomes represent the feature subsets. The fitness of each chromosome is calculated by evaluating the validity of its feature subset, and then the survival probability is determined. According to this probability, the operations of evolution are executed. In this way, new evolved feature subsets are generated. The optimal feature subset is produced by iterating the evolving process. The process of evaluating a feature subset is the most important process in achieving the optimal solution. In this paper, a feature subset is evaluated by the classification error when classifying patterns with the linear decision function that is also generated by GA and K-means algorithm

2.1 Evaluation of Feature Subset Using K-Means Algorithm

To solve the problems of one-stage classifier, the classifier that decide the class of the input pattern by repeating two or more decisions successively, is designed and it is called a multi-stage classifier or decision tree classifier.

It is called 'binary decision tree classifier' that has two child nodes at each node. The binary decision tree classifier divides the patterns into two classes with a suitable feature subset at each node, and this process is iterated until only one pattern class exists in each leaf node.

To select the optimized feature subset, the separability of all the combinations of the features should be evaluated. However, when m components are selected among n components, the number of combinations expressed as nCm becomes a large value even if n and m are not large.

There are some searching algorithms which avoid the exhaustive searching like above, which are top-down, bottom-up, branch and bound, and so on. The feature selection problem can be regarded as an optimization problem. So in this paper, firstly, feature selection is executed using K-means algorithm.

It is based on the minimization of a performance index that is defined as the sum of the squared distances from all points in a cluster domain to the cluster center.

2.2 Evaluation of Feature Subset and Determination of the Linear Decision Function Using GA

The following method is used to minimize the classification error using GA:
Suppose that the given data set is $X = \{\mathbf{x}_{1,}\mathbf{x}_2,\ldots,\mathbf{x}_N\}$ ($\mathbf{x}_k \in R^n$ is the number of features), and $l(j)$ and $r(j)$ are defined as the minimum and maximum values of the j-th feature.

$$l(j) = \min_i x_{ij} \qquad (1)$$
$$r(j) = \max_i x_{ij}$$

In the case of 2-dimensional space, j can have the value of 1 or 2 and on the basis of $l(j)$ and $r(j)$, a rectangle can be constructed that can include all data. Inside the rectangle, two points can be selected arbitrarily, connected by a line. From the coefficients of the line function, a 2-dimensional decision function can be obtained as follows.

$$d(\mathbf{x}) = w_1x_1 + w_2x_2 + w_3 = 0 \qquad (2)$$

When expanding the 2-dimensional case to the n-dimensional case, a hyperplane can be formed by selecting ***n*** points in the hyperspace. And values of $w_1, w_2, \ldots, w_{n+1}$ can be found which appear in the linear decision function of the n-dimensional space, denoted by Eq. (3).

$$\begin{aligned} d(\mathbf{x}) &= w_1x_1 + w_2x_2 + \ldots + w_nx_n + w_{n+1} \\ &= \mathbf{w}_0{}'\mathbf{x} + w_{n+1} \end{aligned} \qquad (3)$$

When matching this concept with a binary string of the GA, ***n*** segments of a binary string indicate one point in the n-dimensional space. In the n-dimensional case, ***n*** points should be selected in such a way that a string is composed of n^2 segments.

Supposing that the length of each segment is *m*-bit, then the total length of the binary string becomes $n^2 m$-bit.

GA determines the decision function that minimizes classification error in a given feature space. Since the minimized error varies with the combination of features, the fitness function is constructed to give high fitness for a combination with a small classification error and low fitness for a combination with a large classification error.

2.3 Construction of the Binary Decision Tree and the Final Recognizer

Using the method described above, a certain feature subset minimizing classification error is chosen. And patterns are classified into two groups at each node with this feature subset. The binary decision tree is constructed by iterating this process until all the classes of patterns appear independently at each leaf node. Because the binary decision tree is constructed for multiple classes rather than just for two it is better to maintain uniform distribution for two separated groups at each node, which means it is better that two separated groups have similar numbers of classes without partiality.

To quantify this, a balance coefficient is defined using the mean and deviation of classes of a new group, as Eq. (4). If the number of patterns of the two separated groups is similar, the balance coefficients are smaller. In this case, because the depth of the binary tree becomes small, the matching time required for recognizing a pattern decreases. The smallest value of the balance coefficient is ***0*** and the largest value is $\sqrt{2}$ for the binary tree case.

$$balance = \sqrt{\frac{\sum_{j=1}^{h} (N_j - \frac{N}{h})^2}{(\frac{N}{h})^2}} \tag{4}$$

In Eq. (4), h is the number of nodes, N is the number of input patterns, and N_j is the number of the patterns included the j-th node. In this paper, a binary tree is constructed, so h becomes ***2***. The fitness function that includes the balance coefficient is defined as.

$$fitness = \frac{1}{1 + w_e \cdot error + w_b \cdot balance} \tag{5}$$

In Eq. (5), *error* and *balance* are the classification error and the balance coefficient between groups, respectively, and w_e and w_b are the weights for weighting each parameter. If both the classification error and the balance coefficient have the value ***0***, fitness has the largest value ***1.*** And the result of the constructed tree can be varied by adjusting of the weights w_e and w_b. For example, if a large value is assigned to w_b, the probability that a more balanced tree structure can be obtained becomes high, while the error rate also becomes high.

After the construction of the binary decision tree, by training BP neural network with the feature subset selected optimally at each node, the final binary tree structured recognizer is realized.

3 Classification of the Defects of Cold Mill Strip Using Binary Tree Classifier

3.1 Extraction of the Features of the Defect Pattern

The defect patterns of cold mill strips can be classified into seven classes: Dull, Oil-drop, Slip, Dent, Scale, Dirt, and Scratch. After preprocessing for the acquired image, we extract six candidate features[1] .

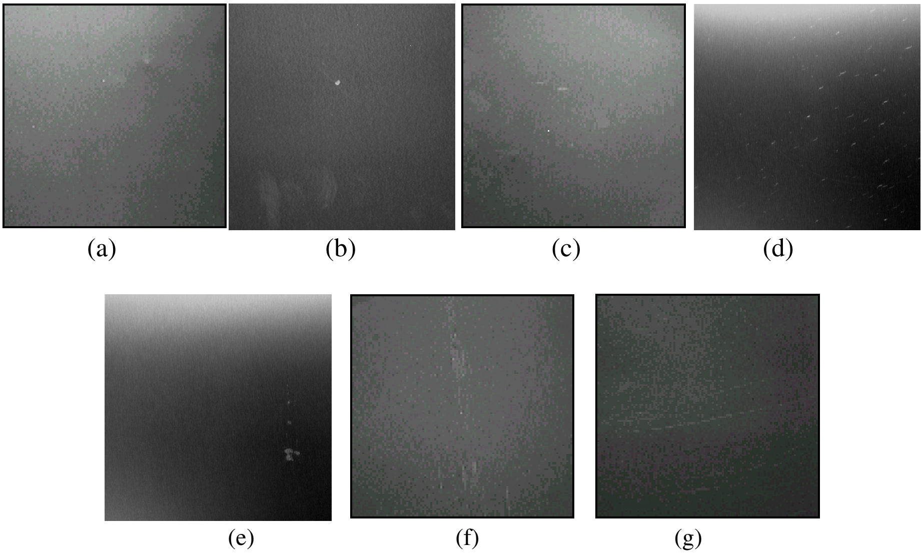


Fig. 2. Defect parrerns of cold mill strip (a) dull (b) oil drop (c) slip (d) dent (e) dirt (f) scale (g) scratch

In this paper, geometrical features are selected as candidate features. They are area, area ratio, and compactness. They are not related to the size and direction of the patterns.

1. *def_area* : the area of a pattern
   (the number of pixels of a defect pattern)

2. *area_ratio* : the ratio of def_area to box_area

$$(area\_ratio = def\_area / box\_area)$$

where *box_area* is the area of the smallest rectangle enclosing the defect pattern.

3. *compactness* : the compactness of a pattern

$$((4\ {*}area)/perimeter^2)$$

where perimeter is the length of the outline.

The probabilistic concept of moment has been used widely in pattern recognition as a practical method to extract features of the shape. Among the features of moment, the information useful for the inspection of the defects of cold mill strips are (a) the length of the longest axis, (b) the ratio of the longest axis to the shortest axis, and (c) the spread of a pattern. These features can be calculated from Eqs. (6)•(10).

4. *length of the longest and shortest axes of a pattern*

$$\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y}) f(x,y) \tag{6}$$

$$a = 2\sqrt{2}\sqrt{(\mu_{20} + \mu_{02} + \sqrt{((\mu_{20} + \mu_{02})^2 + 4\mu_{11}{}^2}}\ \text{: longest} \tag{7}$$

$$b = 2\sqrt{2}\sqrt{(\mu_{20} + \mu_{02} - \sqrt{((\mu_{20} + \mu_{02})^2 + 4\mu_{11}{}^2}}\ \text{: shortest} \tag{8}$$

where $f(x,y)$ is the function of gray level of an image and μ_{ij} is the central moment of a pattern.

5. *ratio of the longest axis to the shortest axis of a pattern*

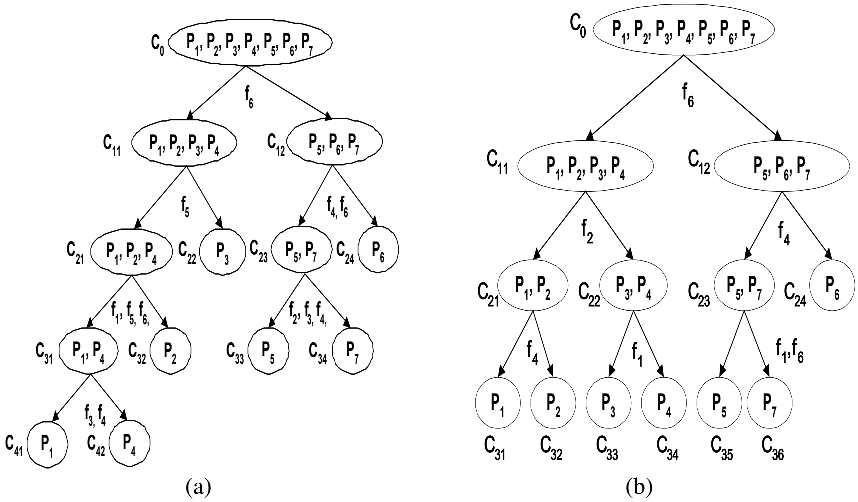
$$axis\_ratio = \frac{b}{a} \tag{9}$$

6. *Spread*

$$spread = \frac{\mu_{02} + \mu_{20}}{\mu_{00}{}^2} \tag{10}$$

3.2 Construction of the Binary Tree Recognizer

The data used in constructing the binary tree recognizer are the feature vectors extracted from the seven types of standard defect patterns. In constructing the binary tree using GA, the weights in Eq. (5), w_e and w_b, are set to 1. Fig. 3(a) shows the binary decision tree constructed from standard patterns by K-means algorithm. Fig. 3(b) represents the binary decision tree by GA. In Fig. 3, P_i is a type of pattern, f_k is a feature, and C_{mn} represents a class at each node.



Symbol	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Pattern	dull	oil	slip	dent	scale	dirt	Scratch

Symbol	f_1	f_2	f_3	f_4	f_5	f_6
Feature	def_area	area_ratio	compactness	axis_ratio	spread	long_axis

Fig. 3. Binary decision tree constructed with standard patterns (a) Evaluation of feature subset by K-means algorithm (b) Evaluation of feature subset by GA

Table 1, 2 show the classification errors, balance coefficients, and the fitness values at each node. The classification errors in Table 1, 2 represent the number of patterns that leave their class when the patterns are divided into two groups at the node.

At each node constructed above, the final recognizer is made by training the BP neural network with the selected feature subset. The number of nodes in the input layer is set to the number of the selected features, and the number of nodes in the hidden layer is set to 10. By setting the number of nodes in the output layer to 2, the output layer represents the binary decision.

Table 1. The patterns and fitness at each node (K-means algorithm)

Node	*Pattern*	*Error/Patterns*	*Feature*	*Fitness*
C_0	$P_1P_2P_3P_4P_5P_6P_7$	0/38	f_6	0.5833
C_{11}	$P_1P_2P_3P_4$	0/15	f_5	0.4926
C_{12}	$P_5P_6P_7$	2/23	f_4f_6	0.2494
C_{21}	$P_1P_2P_4$	0/10	$f_1f_5f_6$	0.5858
C_{23}	P_5P_7	1/8	$f_2f_3f_6$	0.7795
C_{31}	P_1P_4	4/21	f_3f_4	0.1874

Table 2. The patterns and fitness at each node (GA)

Node	*Pattern*	*Error/Patterns*	*Feature*	*Fitness*
C_0	$P_1P_2P_3P_4P_5P_6P_7$	0/38	f_6	0.7180
C_{11}	$P_1P_2P_3P_4$	0/15	f_2	0.5677
C_{12}	$P_5P_6P_7$	2/23	f_4	0.2550
C_{21}	P_1P_2	0/8	f_4	0.5858
C_{22}	P_3P_4	0/7	f_1	0.7795
C_{23}	P_5P_7	3/21	f_1f_6	0.1874

Table 3 shows the results of recognizing the defect patterns of a cold mill strip using the binary tree recognizer.

In Table 3, the recognition rates of Dent and Slip are very low. However, Table 2 shows that the linear classification errors are zero at nodes C_0, C_{11}, and C_{22} when constructing the binary decision tree. This means that the standard patterns of Dent and Slip are classified linearly. Because the least number of features that fit to classify standard patterns are selected, if the number of standard patterns is small, the recognizer becomes sensitive to noise.

Table 3. Recognition rate of each defect pattern

	K-means algorithm		*GA*	
Patterns	No. of recog. / No. of patterns	Recognition rate(%)	No. of recog. / No. of patterns	Recognition rate(%)
Dent	0/3	0	0/3	0
Dull	7/12	58.4	6/12	50
Oil drop	4/4	100	4/4	100
Slip	3/4	75	1/4	25
Dirt	2/2	100	2/2	100
Scale	16/22	72.7	19/22	86.3
Scratch	5/8	62.6	7/8	87.5
Total	37/55	67.2%	39/55	71%

4 Conclusions

In this paper, we used a binary decision tree classifier to recognize the defect patterns of a cold mill strip. We have used the cold mill strip of POSCO(Pohang Steel Company), which consists of 55 defect patterns. At each node of the binary tree, K-means and GA were used for the selection of the best feature subset and the linear decision function. There are two advantages of this method. One is that the construction of the binary decision tree and the selection of the best feature subset can be executed automatically for the given patterns. The other is that by designing the fitness function of GA properly, the decision tree can be obtained by considering the balance of the classes as well as the classification error.

In this experiment, GA is better than K-means algorithm in performance. But current performance is about 71% of recognition rate. Further studies should be made to design classifiers which have more generalization capabilities and feature extraction methods which are mutual helpful for the recognition of the defect pattern of a cold millstrip.

References

1. Kim K.M.: Design of a Binary Decision Tree for Recognition of the Defect Patterns of Cold Mill Strip Using Generic Algorithm. Proc. AFSS. (1998) 208-212
2. Brill F., Brown D., Martin W.: Fast Genetic Selection of Features for Neural Network Classifiers. IEEE Trans. Neural Networks, 32 (1992) 324-328
3. Yao L.: Nonparametric Learning of Decision Regions via the Genetic Algorithm. IEEE Trans. Sys. Man Cybern., 26 (1996) 313-321
4. Mui J.K., Fu K.S.: Automated Classification of Nucleated Blood Cells using a Binary Tree Classifier. IEEE Trans. Pattern Analysis Machine Intelligence, 5 (1980) 429-443
5. Safavian S.R., Landgrebe D.: A Survey of Decision Tree Classifier Methodology. IEEE Trans. Sys. Man Cybern., 21 (1991) 660-674
6. Payne H.J., Meisel W.S.: An Algorithm for Constructing Optimal Binary Decision Trees. IEEE Trans. Computers, 26 (1997) 905-916
7. Swain P.H., Hauska H.: The Decision Tree Classifier: Design and Potential. IEEE Trans. Geosci. Elec., 15 (1997) 142-147
8. Jung S.W., Park G.T.: Design and Application of Binary Decision Tree using Genetic Algorithm. Journal of the KITE, 33(6) (1996) 1122-1130

Evolutionary RSA-Based Cryptographic Hardware Using the Co-design Methodology

Nadia Nedjah and Luiza de Macedo Mourelle

Department of System Engineering and Computation, Faculty of Engineering,
State University of Rio de Janeiro,
Rio de Janeiro, Brazil
{nadia, ldmm}@eng.uerj.br
http://www.eng.uerj.br /~ldmm

Abstract. Most cryptography systems are based on the modular exponentiation to perform the non-linear scrambling operation of data. It is performed using successive modular multiplications, which are time consuming for large operands. Accelerating cryptography needs optimising the time consumed by a single modular multiplication and/or reducing the total number of modular multiplications performed. Using a genetic algorithm, we first yield the minimal sequence of powers, generally called addition chain, that need to be computed to finally obtain the modular exponentiation result. Then, we exploit the co-design methodology to engineer a cryptographic device that accelerates the encryption/decryption throughput without requiring considerable hardware area.

1 Introduction

The modular exponentiation is a common operation for scrambling and is used by several public-key cryptosystems, such as the RSA encryption scheme [1]. It consists of a repetition of modular multiplications: $C = T^E \bmod M$, where T is the plain text such that $0 \leq T < M$ and C is the cipher text or vice-versa, E is either the public or the private key depending on whether T is the plain or the cipher text, and M is called the modulus. The decryption and encryption operations are performed using the same procedure, i.e. using the modular exponentiation.

The performance of such cryptosystems is primarily determined by the implementation efficiency of the modular multiplication and exponentiation. As the operands, i.e. the plain text of a message or the ciphertext (possibly a partially ciphered) are usually large (i.e. 1024 bits or more), and in order to improve time requirements of the encryption/decryption operations, it is essential to attempt to minimise the number of modular multiplications performed as well as the time needed to perform a single modular multiplication.

Most of the work [2], [3], [4] on improving the characteristics, i.e. encryption/ decryption throughput and required resources, focus on one aspect: minimising the exponentiation time by implementing the operation on hardware. However, this solution requires a lot of hardware area. In this paper, we propose and implement a novel

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 351-360, 2004.

solution that minimises the number of required modular multiplications along with the modular multiplication time without too much increase in resource requirements. We do so using genetic algorithms [5] and the co-design methodology [6]. The solution proposed here finds a balance between the two requirements: time and area. Furthermore, it allows one to change the encryption and decryption key freely without any extra cost.

First, we introduce the concept of evolutionary addition chains as well as addition chain based methods to perform modular exponentiation. Then, we introduce Montgomery's Algorithm used to implement the modular multiplication. Thereafter, we describe the co-design system. Consequently, we discuss the architecture used to implement the mixed solution. Finally, we draw some conclusions based on the analysis of the system developed.

2 Evolutionary Addition Chains

It is clear that one should not compute T^E then reduce the result modulo M as the space requirements to store T^E is $E \times \log_2 M$, which is huge. A simple procedure to compute $C = T^E \bmod M$ is based on the paper-and-pencil method. This method requires $\tilde{E}1$ modular multiplications computing all powers of T: $T \rightarrow T^2 \rightarrow \ldots \rightarrow T^{\tilde{E}1} \rightarrow T^E$. The paper-and-pencil method computes more multiplications than necessary. For instance, to compute T^8, it needs 7 multiplications, i.e. $T \rightarrow T^2 \rightarrow T^3 \rightarrow T^4 \rightarrow T^5 \rightarrow T^6 \rightarrow T^7 \rightarrow T^8$. However, T^8 can be computed using only 3 multiplications $T \rightarrow T^2 \rightarrow T^4 \rightarrow T^8$. The basic question is: what is the fewest number of multiplications to compute T^E, given that the only operation allowed is multiplying two already computed powers of T? Answering the above question is *NP*-hard, but there are several efficient algorithms that can find a near optimal one.

The addition chain based methods attempt to find a chain of numbers such that the first number of the chain is 1 and the last is the exponent E, and in which each member of the chain is the sum of two previous members. For instance, the longest addition chain is [1, 2, 3, …, E-2, E-1, E]. An *addition chain* of length l for an integer n is a sequence of integers $[a_0, a_1, a_2, \ldots, a_l]$ such that $a_0 = 1$, $a_l = n$ and $a_k = a_i + a_j$, $0 \le i \le j < k \le l$. The algorithm used to compute the modular exponentiation $C = T^E \bmod M$, is specified by Algorithm 1.

```
Algorithm 1. AdditionChainBasedMethod(T, M, E)
let [$a_0$=1 $a_1$ $a_2$ … $a_l$=E] be an addition chain for E;
powers[0] = T;
for k := 1 to l
   let $a_k$ = $a_i$ + $a_j$ | i<k and j<k;
   powers[k] := powers[i] × powers[j] mod M;
return powers[l];
End.
```

Computing the minimal addition chain for a given exponent is a hard problem [5], [7]. We used genetic algorithms [8] to yield optimal addition chains for large exponents [5]. We showed that the addition chains obtained using the evolutionary methodology are always very much better than those used by the traditional exponentiation methods such as the m-ary methods and sliding window methods [9].

3 Montgomery's Algorithm

One of the widely used algorithms for efficient modular multiplication is Montgomery's algorithm [10]. This algorithm computes the product of two integers modulo a third one without performing division by *M*. It yields the reduced product using a series of additions

Let *A*, *B* and *M* be the multiplicand, the multiplier and the modulus respectively and let *n* be the number of digits in their binary representation, i.e. the radix is 2. So, we denote *A*, *B* and *M* as follows:

$$A=\sum_{i=0}^{n-1} a_i \times 2^i,\quad B=\sum_{i=0}^{n-1} b_i \times 2^i \text{ and } M=\sum_{i=0}^{n-1} m_i \times 2^i$$

The pre-conditions of the Montgomery algorithm are as follows:

- The modulus *M* needs to be relatively prime to the *radix*, i.e. there exists no common divisor for *M* and the radix;
- The multiplicand and the multiplier need to be smaller than *M*.

As we use the binary representation of the operands, then the modulus *M* needs to be odd to satisfy the first pre-condition.

The Montgomery's algorithm uses the least significant digit of the accumulating *modular partial product* to determine the multiple of *M* to subtract. The usual multiplication order is reversed by choosing multiplier digits from least to most significant and shifting down. If *R* is the current modular partial product, then *q* is chosen so that $R + q \times M$ is a multiple of the radix *r*, and this is right-shifted by *r* positions, i.e. divided by *r* for use in the next iteration. So, after *n* iterations, the result obtained is $R = A \times B \times r^n \bmod M$. A modified version of the Montgomery's algorithm is given in Algorithm 2.

```
Algorithm 2. MontgomeryAlgorithm(A, B, M)
int R := 0;
for i := 0 to n-1
  R := R + $a_i \times$ B;
  if $r_0$ = 0 then R := R div 2
  else R := (R + M) div 2;
return R;
End.
```

In order to yield the right result, we need an extra Montgomery modular multiplication by the constant 2^{2n} mod *M*. However, as the main objective of the use of Montgomery modular multiplication algorithm is to compute exponentiations, it is preferable to Montgomery pre-multiply the operands by 2^{2n} and Montgomery post-multiply the result by 1 to get rid of the 2^n factor. Now, we concentrate on describing the implementation of the Montgomery multiplication algorithm.

4 The Co-design Architecture

Our investigation is based on Algorithm 1, assuming that the addition chain is provided. The software approach consists of implementing the algorithm in a programming language, such as C, and executing the compiled code in a general-purpose computer.

The bottleneck in the software approach is the evaluation of the modular multiplication. Therefore, we decided to move this computation to hardware in order to explore the speedup that can be achieved by a hardware implementation. From this point on, we will have a mixed implementation, in which part of the initial specification is in software and another part is in hardware. Consequently, we will have to deal with the interaction between these two subsystems. The dynamics within the co-encryption/decryption system is described in Fig. 1.

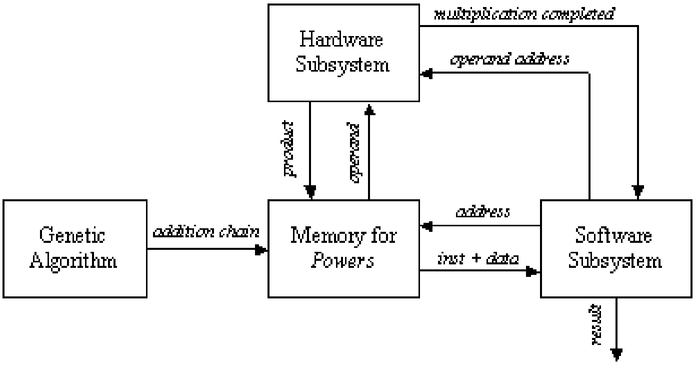


Fig. 1. Dynamics within the mixed encryption/decryption process

The execution cycle within the co-design system is described in the following seven steps:

1. The genetic algorithm evolves a minimal addition chain for the given encryption/decryption key;
2. The evolutionary addition chain is stored into the co-system shared memory;
3. The software subsystem executes a program that implements the computation of Algorithm 1 and is stored in the shared memory;
4. The software subsystem finds the operands of the modular multiplication the hardware subsystem has to perform;
5. The software subsystem notifies the hardware subsystem to start the modular multiplication and waits;
6. Once the modular product is reached, the hardware subsystem notifies the software subsystem and halts;
7. The software subsystem checks whether the last multiplication was performed; If yes, it reads the shared memory to acquire the result of the modular exponentiation, otherwise it performs step 4 repeatedly.

In the following sections, we explain in details, the architecture of each of the subsystems.

4.1 The Genetic Algorithm

Genetic algorithms [8] maintain a *population* of *individuals* that evolve according to *selection* rules and other *genetic operators*, such as *mutation* and *recombination*. Each individual receives a measure of *fitness*. *Selection* focuses on high fitness individuals. Mutation and recombination provide general heuristics that simulate the reproduction or *crossover* process. Those operators attempt to perturb the characteristics of the parent individuals as to generate *distinct* offspring individuals.

The addition chain minimisation problem consists of finding a sequence of numbers that constitutes an addition chain for a given exponent. The sequence of numbers should be of a minimal length.

Encoding of individuals is one of the implementation decisions one has to take in order to use genetic algorithms. It very depends on the nature of the problem to solve. There are several representations that have been used with success: *binary encoding* which is the most common mainly because it was used in the first works on genetic algorithms, represents an individual as a string of bits; *permutation encoding*, mainly used in ordering problem, encodes an individual as a sequence of integers; *value encoding* represents an individual as a sequence of values that consist of an evaluation of some aspect of the problem [7], [8].

In our implementation, an individual represents an evolutionary addition chain. We use the binary encoding wherein 1 implies that the entry number is a member of the addition chain and 0 otherwise. Let $n = 9$ be the exponent. The encoding of Fig. 2 represents the addition chain [1, 2, 4, 5, 9]:

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
1	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**1**

Fig. 2. Addition chain encoding

4.2 Software Subsystem Architecture

In Algorithm 2, the formal parameters can be of 1024 bits. Therefore, instead of passing these values, we decided to pass the indexes to the array *powers* (*i*, *j* and *k*), together with the pointer to *powers* address of *M* and that of *powers*. Algorithm 3 below shows the modified version of Algorithm 1.

```
Algorithm 3. ModAdditionChainBasedMethod(T, M, E)
let [$a_0$=1,$a_1$,$a_2$,...,$a_l$=E] be an addition chain for E;
powers[0] := T;
for k := 1 to l
  find k | i<k and j<k, $a_k$ = $a_i$ + $a_j$;
  ModifiedMontgomery(i, j, k, M, powers, size);
return powers[l];
End.
```

In order to perform the chosen computation, the hardware subsystem needs the function's parameters, which are sent by the software subsystem. Integer and pointer parameters are passed via memory-mapped registers, while data arrays are stored in the shared memory. Algorithm 2 must be modified as well, so as to include the necessary hardware interaction, which can be seen in Algorithm 4 below:

```
Algorithm 4. ModifiedMontgomery(i,j,k,&M,&powers, size)
    char* const parameter$_0$ := (char*) 0xF000;
    char* const parameter$_1$ := (char*) 0xE000;
    char* const parameter$_2$ := (char*) 0xD000;
    char** const parameter$_3$ := (char**) 0xC000;
    char** const parameter$_4$ := (char**) 0xB000;
    *parameter$_0$ := i; *parameter$_1$ := j;
    *parameter$_2$ := k;
    if k = 1 then
      *parameter$_3$ := &M;
      *parameter$_4$ := &powers;
      *parameter$_5$ := size;
     start();
     waitForInterruption();
     acknowledge();
End.
```

As can be seen from Algorithm 4, parameter0, parameter1, parameter2, parameter3, parameter4 and parameter5 contain the addresses of the parameter registers located in the hardware subsystem. After their initialisation, the hardware subsystem can be started to execute the computation. In our case, parameters i, j and k are used to address the elements of the array powers, while parameter powers holds the address of the first element of the corresponding array. Hence, i, j and k are used as displacement within the array area. Since M can be large, we decided to keep M in the shared memory and pass its address only. Notice that it is up to the hardware subsystem to get the necessary data from the shared memory, once it is started. The software subsystem, then, waits for an interrupt from the hardware subsystem, indicating it has completed the operation.

4.3 Hardware Subsystem Architecture

The hardware subsystem comprises the hardware function and the interface logic. The latter deals with the communication between the hardware subsystem and the other entities, i.e. software subsystem and the shared memory. The characteristics of the interface depend closely on the implementation platform. Therefore, we will deal with it in the next section.

The hardware function computes the modular product of two given operands using Montgomery's algorithm described in Section 3. Fig. 3 shows the architecture of an iterative implementation [4] for the Montgomery modular multiplication method [10]. The values of A and B are obtained from the memory, where the array elements are stored, using parameters i and j, respectively. These indexes are provided by the software subsystem. The obtained modular product is stored in the same array powers in entry k = i + j.

The first multiplexer of the proposed architecture, i.e. mux21, passes 0 or the content of register B depending on whether bit a0 indicates 0 or 1 respectively. The second multiplexer, i.e. mux22 passes 0 or the content of register M depending on whether bit r0 indicates 0 or 1 respectively. The first adder, i.e. adder1, delivers the sum R + ai × B (line 2 of Algorithm 2), and the second adder, i.e. adder2, yields the sum R + M (line 4 of the same algorithm). The shift register shift register1 provides the bit ai. At each iteration i of the multiplier, this shift register is right-shifted once, so that the least significant bit of shift register1 contains ai.

The role of the controller consists of loading A, B and M and synchronising the shifting and loading operations of shiftregister1 and shiftregister2, and controlling the number of necessary iterations. Furthermore, embedded into the controller hardware, we find the steps for parameter passing as well as the handshake protocol between the hardware and software subsystems. The handshake control register allows yielding the start and done commands from the software and hardware subsystems respectively.

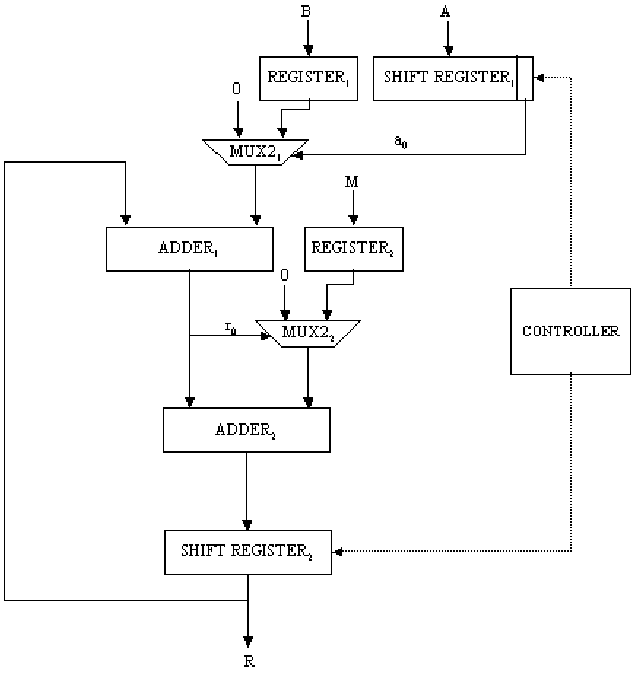


Fig. 3. Montgomery multiplication hardware

In order to synchronise the work of the components of the architecture, the controller is implemented as a state machine, which has 6 states defined as follows:

Memory read operations (to obtain the values of A, B and M) as well as memory write operations (to store the modular products) are embedded in the specification of the hardware subsystem and performed by the interface logic.

The interface between the hardware function and the software subsystem uses a control register CR through which a handshake protocol is implemented. When the software subsystem wants to call the hardware function, it asserts the start bit of CR (line 11 in Algorithm 4). When the hardware function completes the execution, it asserts the done bit of CR. When the software subsystem acknowledges the end of the hardware function operation (line 13 in Algorithm 4), it withdraws the start command by resetting the start bit of CR. When the interface logic detects that the start bit was reset, it resets the done bit, thus completing the handshake.

```
$S_0$: Initialise state machine;
$S_1$: If start = 0 then Go to $S_2$ Else Go to $S_1$;
$S_2$: done := 0;
     If start = 1 then Go to $S_4$
     Else If parameters = 0 then Go to $S_2$;
$S_3$: If $parameter_0$ then Load i into $REGISTER_i$
     Else If $parameter_1$ then Load j into $REGISTER_j$
          Else If $parameter_2$ then Load k into $REGISTER_k$
               Else If $parameter_3$ then Load &M into $REGISTER_M$
                    Else If $parameter_4$ then
                           Load &powers into $REGISTER_P$;
                         Else If $parameter_4$ then
                              Load sise into counter;
     Go to $S_2$;
$S_4$: Load powers[i] from memory into SHIFT $REGISTER_1$;
$S_5$: Load powers[j] from memory into $REGISTER_1$;
$S_6$: If k = 1 then
          Load M from memory into $REGISTER_2$;
$S_7$: Wait for $ADDER_1$; Wait for $ADDER_2$;
     Decrement counter;
$S_8$: Load partial result into SHIFT $REGISTER_2$;
$S_9$: Enable SHIFT $REGISTER_2$; Enable SHIFT $REGISTER_1$;
     If counter = 0 then Go to $S_{10}$ Else Go to $S_7$;
$S_{10}$: Load SHIFT $REGISTER_2$ into memory powers[k];
     done := 1; Go to $S_1$
```

5 Implementation Platform

In order to obtain a final implementation, we need a processor capable of executing the software instructions (software subsystem) and a hardware device capable of executing the chosen computation (hardware subsystem). Our co-design platform consists of the XS40 board, from Xess [11], which is based on the Intel 80C31 micro-controller, the XilinxTM XC4010XL FPGA [12] and 32KB of SRAM, shared by the hardware and the software subsystems. A simplified version of the co-design architecture is seen in Fig. 4 and the XS40 co-design board can be found in [11].

While the hardware subsystem is computing the required modular product (computation of line 4 in Algorithm 3), the micro-controller finds the entries of array powers in which operands of the next modular multiplications (computation of line 3 in Algorithm 3) are located. Interleaving the work of the hardware function with that of the micro-controller improves a great deal the overall performance of the encryption/decryption co-design system.

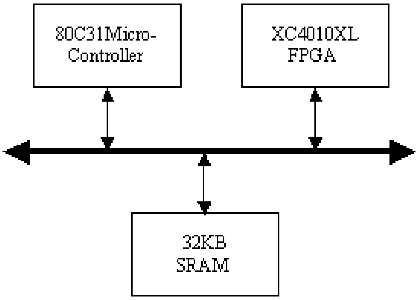


Fig. 4. Codesign system architecture

6 Timing and Area Characteristics

In this section, we compare the evolutionary cryptographic hardware, which is a mixed system (i.e. software and hardware) described throughout this paper with the software-only and hardware-only versions. The software-only system is implemented in asm51 assembly language [13]. Recall that the software subsystem of the proposed solution is also implemented using asm51. The software subsystem of the mixed system proposed is programmed using this same language. The hardware-only system is implemented into xs4000. The hardware subsystem of the mixed system is also implemented into the same fpga family.

The software-only and the hardware-only implementations are based on the binary exponentiation. The latter implementation was developed by the authors in [4]. Table 1 shows the hardware area and response time for the three alternative implementations as well as the area×time factor.

Table 1. Hardware area (CLBs), response time (ns) and performance factor under the three implementations: software-only, hardware-only and mixed system, for operand sizes 512 and 1024

Requirements / *System*	**Hardware area**		**Response time**		**Area × Time**	
	512	1024	512	1024	512	1024
Software-only	–	–	1982	3491	–	–
Hardware-only	811	1679	713	1354	578243	2273366
Mixed	195	431	1029	1782	200655	768042

7 Conclusion

In this paper, we proposed and implemented a novel solution that focuses on the two major aspects impacting on the performance of any given cryptosystems based on modular exponentiation as a non-linear function for data scrambling: (i) the proposed solution minimises the number of required modular multiplications and; (ii) the modular multiplication time, without too much increase in resource requirements. To do so, we evolve, using genetic algorithms, a minimal addition chain based on which we perform the modular exponentiation. Moreover, we exploited the co-design methodology to partition the modular exponentiation into two subsystems: the hardware subsystem and the software subsystem. Given the adequate operands, the former performs a single modular multiplication. The latter coordinates the work of the hardware subsystem based on the evolutionary addition chain.

The solution proposed and implemented finds a balance between the two requirements: time and area. Furthermore, it allows one to change of the encryption and decryption key freely without any extra cost. We demonstrated that the response time of the mixed implementation is not that bad with respect to that of the hardware-only implementation. As a matter of fact, the co-design based implementation is about 27% slower than the hardware-only one. However, the mixed implementation requires very much less hardware than the hardware-only solution. The latter consumes about four times more hardware area that the former. Finally, we showed that the co-design based system improves considerably in about 65%.

References

1. Rivest, R.L., Shamir, A. and Adleman, L., *A method for obtaining digital signature and public-key cryptosystems*, Communication of ACM, vol. 21, no.2, pp. 120-126, 1978.
2. Blum, T. and Paar C., *Montgomery modular exponentiation on reconfigurable hardware*, Proceedings of the 14th. IEEE Symposium on Computer Arithmetic, Australia, 1999.
3. C. D. Walter, *Systolic modular multiplication*, IEEE Transactions on Computers, 42(3):376-378, 1993.
4. Nedjah, N and Mourelle, L.M., *Two hardware implementations for the Montgomery multiplication: sequential vs. parallel*, Proceedings of the 15th. Symposium on Integrated Circuits and Systems Design, Brazil, IEEE Computer Society, pp. 3-8, 2002.
5. Nedjah, N. and Mourelle, L.M., *Minimal addition chains for efficient modular exponentiation using genetic algorithms*, Proceedings of the Fifteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Cairns, Australia, Lecture Notes in Computer Science, Springer-Verlag, vol. 2358, pp. 88-98, 2002.
6. F. Balarin et al., *Hardware-software co-design of embedded systems: the polis approach*, Kluwer Academic Publishers, 1997.
7. DeJong, K. and Spears, W.M., *Using genetic algorithms to solve NP-complete problems*, Proceedings of the Third International Conference on Genetic Algorithms, pp. 124-132, Morgan Kaufmann, 1989.
8. Haupt, R.L. and Haupt, S.E., *Practical genetic algorithms*, John Wiley and Sons, 1998.
9. Nedjah, N. and Mourelle, L.M., *Efficient parallel modular exponentiation algorithm*, Proceedings of the Second International Conference on Information Systems, Izmir, Turkey, Lecture Notes in Computer Science, vol. 2457, pp. 405-414, 2002.
10. P.L. Montgomery, *Modular Multiplication without trial division*, Mathematics of Computation 44, pp. 519-521, 1985.
11. Xess, http://www.xess.com, 2003.
12. Intel, *MCS^{TM}51 family of micro-controllers architectural overview*, htt://www.intel.com, 2003
13. Xilinx, http://www.xilinx.com, 2003.

GA-EDA: Hybrid Evolutionary Algorithm Using Genetic and Estimation of Distribution Algorithms

J.M. Pe–a, V. Robles, P. Larra–aga, V. Herves, F. Rosales, and M.S. P□rez

Universidad Polit□cnica de Madrid, Madrid, Spain
{jmpena, vrobles, vherves, frosal, mperez}@fi.upm.es
Universidad del Pa’s Vasco, San Sebasti□n, Spain
ccplamup@si.ehu.es

Abstract. Evolutionary techniques are one of the most successful paradigms in the field of optimization. In this paper we present a new approach, named GA-EDA, which is a new hybrid algorithm based on genetic and estimation of distribution algorithms. The original objective is to get benefits from both approaches. In order to perform an evaluation of this new approach a selection of synthetic optimizations problems have been proposed together with two real-world cases. Experimental results show the correctness of our new approach.

Keywords. Genetic Algorithms and Heuristic Search

1 Introduction

Evolutionary techniques stand from the assumption that a restricted set of solutions could be evolved to improve solutions in an iterative process. The evolutionary process is driven by a fitness function, which measures how good each solution is.

Many pure (GAs and EDAs) techniques have been proposed as well as other combination of them, named hybrid algorithms. Examples of them are: ERA, which incorporates, simulated annealing. [Rodriguez-Tello & Torres-Jimenez, 2003]; GASAT that incorporates local search within the genetic framework [Hao & Lardeux & Saubion, 2002]; and an integrated Genetic Algorithm with Hill Climbing that solves the matrix bandwidth minimization problem [Lim & Rodrigues & Xiao, 2003]. Also, a hybrid algorithm based on the combination of EDA with Guided Local Search (GLS) for Quadratic Assignment Problems (QAP) [Zhang & Sun & Tsang & Ford, 2003]; another hybrid genetic algorithm that combines efficient local heuristic and aging mechanism for the hexagonal tortoise problem [Choe & Choi & Moon, 2003].

In this paper we present a new approach, named GA-EDA, which is a new hybrid algorithm based on genetic and estimation of distribution algorithms.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 361-371, 2004.

2 Evolutionary Optimization Methods

Among the different evolutionary techniques the best known are Genetic Algorithms, although new approaches, like Estimation of Distribution Algorithms, have arise in the very last years.

2.1 Genetic Algorithms

Genetic Algorithms are heuristics search and optimization algorithms, highly parallel, inspired by the Darwinian principle of natural selection and genetic reproduction [Goldberg, 1989].

Genetic Algorithms begins with a population of individuals, each one representing a possible solution of a given problem. These individuals are represented as chromosomes. Chromosomes generally are sequences of bits, but often the problem demands one more complex representation. Any chosen representation, should to be able to represent the entire space search to investigate. Representation must be minimum since if it contains unnecessary information the size of the space search increase and therefore the efficiency of the GA decreases during the search.

Pseudocode for the GA approach:

```
$P_0 \leftarrow$ Generate $M$ individuals (the initial population)
Repeat until stopping criterion is reached ($i = 1...n$):
    • Selection:
        - $P_{intermediate} \leftarrow$ Select $N$ individuals ($N \leq M$) from $P_{i-1}$ according to
          some selection mechanisms
        - Select $P$ individuals ($P \leq N$) from $P_{intermediate}$ that will be the
          progenitors
    • Reproduction:
        - $P$ individuals of $P_{intermediate}$ are selected and joined in pairs, and $Q$
          descendants are generated
        - Pintermediate $\leftarrow N + Q$
    • Replacement:
        - $Pi \leftarrow$ Select $M$ individuals from $P_{intermediate}$, generally the fittest.
```

2.2 Estimation of Distribution Algorithms

EDAs [Larra–aga & Lozano, 2001] [Mhlenbein, 1998] are non-deterministic, stochastic heuristic search strategies that form part of the evolutionary computation approaches, where a number of solutions or individuals are created every generation, evolving once and again until a satisfactory solution is achieved. In brief, the characteristic that differentiates most EDAs from other evolutionary search strategies such as GAs is that the evolution from a generation to the next one is done by estimating the probability distribution of the fittest individuals, and afterwards by sampling the induced model. This avoids the use of crossing or mutation operators, and the number of parameters that EDAs require is considerably reduced.

In the pseudocode of a generic EDA algorithm, we can distinguish four main steps:

1. At the beginning, the first population D_0 of M individuals is generated.

2. A number N ($N <= M$) of individuals are selected, usually the fittest.

3. The n□dimensional probabilistic graphical model that better expresses the dependencies among the n variables is induced.

4. A new population of M new individuals is obtained by simulating the probability distribution learnt in the previous step.

2.3 Comparative Results

There are several studies of the properties and qualities of these two approaches for different problems (see chapters 13,16 and 17 of [Larra–aga & Lozano, 2001]). One of the most important results obtained on these and similar studies is that none of them outperforms the other for all the possible problems. There are cases in which GAs converge slower to the solution and there are other cases in which EDAs fall in a local optimum. Sometimes the absolute optimum is obtained only by one of these algorithms. The reason depends on characteristics of the very problem, and only for few specially designed problems is possible to predict whether GAs or EDAs are going to perform better.

3 Hybrid GA-EDA Algorithm

On this paper we propose a new algorithm based on both techniques. The original objective is to get benefits from both approaches. The main difference from these two evolutionary strategies is how new individuals are generated. These new individuals generated on each generation are called *offspring*. On one hand, GAs uses crossover and mutation operators as a mechanism to create new individuals from the best individuals of the previous generation. On the other, EDAs builds a probabilistic model with the bests individuals and then sample the model to generate new ones.

Participation Function Our new approach generates two groups of offspring individuals, one generated by the GA mechanism and the other by EDA one. $Population_{p+1}$ is composed by the best overall individuals from (i) the past population ($Population_p$), (ii) the GA-evolved offspring, and (iii) EDA-evolved offspring.

The individuals are selected based on their fitness function. This evolutionary schema is quite similar to Steady State GA in which individuals from one population, with better fitness than new individual from the offspring, survive in the next one. In this case we have two offspring pools. Figure 1 shows how this model works.

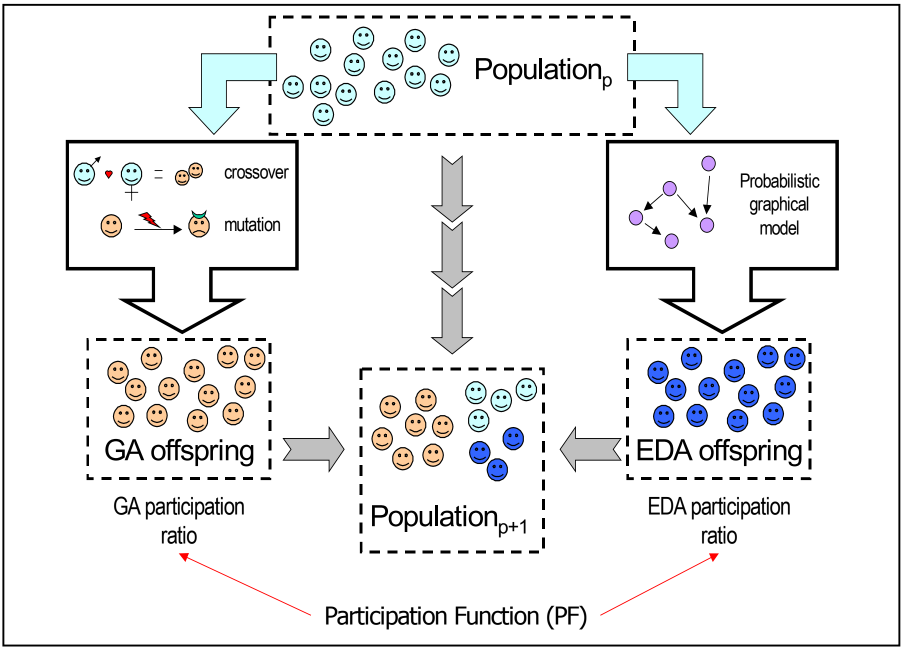


Fig. 1. Hybrid Evolutionary Algorithm Schema

On this new approach an additional parameter appears, this parameter has been called *Participation Function* (PF). PF provides a ratio of how many individuals are generated by each mechanism. In other words, the size of GA and EDA offspring sets. The size of these sets also represents how each of these mechanisms participates on the evolution of the population. These ratios are only a proportion for the number of new individuals each method generates, it is not a proportion of individuals in the next population, which is defined by the quality of each particular individual. If a method were better that the other in terms of how it combines the individuals there would be more individuals from this offspring set than the other.

The following alternatives for Participation Functions are introduced:

Constant Ratio (*x% EDA / y% GA*)

The percentage of individuals generated by each method is constant during all the generations. For example, *30%* of the individuals are generated by GA crossover and mutation and *70%* by the EDA probabilistic graphical model.

Incremental Ratio (*EDA*++ and *GA*++)

The partition ratio for one of the mechanism increases from one generation to the other. There are two incremental Participation Functions, GA Incremental Function and EDA Incremental Function. The ratio is defined by the formula[1]:

$$ratio = \frac{gen}{M + gen}$$

Alternative Ratio (*ALT*)

On each generation it alternates either GA or EDA generation method. If the generation is an even number GA mechanism generates all offspring individuals, if it is an odd number is the EDA method.

Dynamic Ratio (*DYNAMIC*)

As a difference with the previous Participation Functions that are static (and deterministic), we also propose a dynamic adaptative function. The idea is to have a mechanism that increases the participation ratio for the method that happens to generate better individuals. This function is evaluated each generation considering the possibility to change the participation criterion (defined by the `ratio` array).

This function performs according to the following algorithm:

```
diff=(MAX(avg_score[GA],avg_score[EDA])-base)/
     (MIN(avg_score[GA],avg_score[EDA])-base);
if(avg_score[GA]>avg_score[EDA]){
 ratio_inc=ratio[EDA]*ADJUST*diff;
 ratio[GA]  += ratio_inc;        ratio[EDA]= 1.0 - part[GA];
}
else if(avg_score[GA]<avg_score[EDA]){
 ratio_inc=ratio[GA]*ADJUST*diff;
 ratio[EDA] += ratio_inc;        ratio[GA] = 1.0 - part[EDA];
}
```

Where `avg_score` is an array of the average fitness score of the top *25%* of the individual generated by each of the offspring methods[2]. As the best fitness score is monotonically increasing this value is always greater than 1. `ADJUST` is a constant that defines the size of the steps of the dynamic update (*5%* in our experimentation).

This algorithm starts with *50%/50%* ratio distribution between the two methods. On each generation the best offspring individuals from each method are compared and the wining method gets a *5%* of the opposite method ratio (scaled by the amount of relative difference between the methods, `diff` variable). This mechanism provides a contest-based dynamic function, in which methods are competing to get higher ratio as they generate better individuals.

[1] *gen* is the number of the generation and *M* is called the Mid-point that represents at which generation the ratio is *50%/50%*. Function is 0 at the first generation and never reaches 1.

[2] `base` is the best fitness score obtained in the first generation (used to scale fitness values).

4 Evaluation of the New Algorithm

The hybrid algorithm proposed is composed by the simplest versions of both GA and EDA component. In this sense a single bit-string chromosome has been used to code all the problems. GA uses □Roulette Wheel□ selector, one-point crossover, flip mutation and uniform initialization. EDA uses UMDA probabilistic model. The overall algorithms generate an offspring twice the size of the population (this offspring is then divided between two methods depending on the ratios provided by the Participation Function). The composition of the new population is defined by a deterministic method, selecting the best overall fitness scores from the previous population and both offspring sets.

All experiments have been run ten times, and the values in the figures of the experimental result section are the average of these executions. As Participation Functions for the hybrid approach we have tested the next ones: *75% EDA/ 25% GA*, *50% EDA / 50% GA* and *25% EDA / 75%* GA as constant ratio functions, as well as *EDA++*, *GA++*, *ALT*, and *DYNAMIC* functions.

4.1 Description of the Problems

Four classes of problems were empirically tested on our new hybrid approach, two artificial problems (*4-bit fully deceptive function* and *240 bit Holland royal road*) and two real problems (*SAT problem* and *feature subset selection* problem).

4-Bit Fully Deceptive fuNction

Deceptive trap functions are used in many studies of GAs because their difficulty is well understood and it can be regulated easily [Deb & Golberg, 1993]. We have used the 4-bit fully deceptive function of order 2, defined by Whitley and Starkweather in their paper GENITOR II [Whitley & Starkweather, 1990]. The problem is a 40 bit long maximization problem, and is comprised of 10 sub-problems, each 4 bits longs.

240 Bit Holland Royal Road

The Royal Road functions were introduced in [Mitchell et al., 1992]. They were designed as functions that would be simple for a genetic algorithm to optimize, but difficult for a hillclimber. In [Holland, 1993], Holland presented a revised class of Royal Road functions that were designed to create insurmountable difficulties for a wider class of hillclimbers, and yet still admissible to optimization by a GA.

The Holland Royal Road function takes a binary string as input and produces a real value. The function is used to define a search task in which one wants to locale strings that produces high function values. The string is composed of 2^k non-overlapping continuous regions, each of length $b+g$. With Hollands□s defaults, $k=4$, $b=8$, $g=7$, there are 16 regions of length 15, giving and overall string length of 240. Each region is divided into two non-overlapping pieces. The first, of length b, is called the block, and the second, of length g, is called the gap. In the fitness calculation, only the bits in the block part of each region are considered.

SAT Problem

The goal of the satisfiability (SAT) problem [Rodriguez-Tello & Torres-Jimenez, 2003] is to find an assignment of truth-values to the literals of a given boolean formula, in its conjunctive normal form, that satisfies it. In theory SAT is one of the basic core NP-complete problems. In practice, it has become increasingly popular in different research fields, given that several problems can be easily encoded into propositional logic formula such as planning, formal verification, knowledge representation and so on.

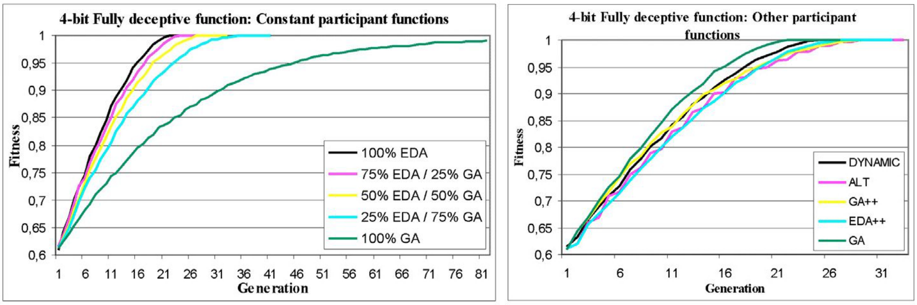
In GAs and EDAs the SAT problem is represented using binary strings of length *n* in which the *i-th* bit represents the truth-value of the *i-th* propositional variable in the formula. The fitness function used is the fraction of clauses satisfied. To test the algorithm developed the SAT instances *4blocksb.cnf* was used, since they are widely-known and easily available from the SATLIB benchmark[3].

Feature Subset Selection

Feature Subset Selection (FSS) is a well-known task in the Machine Learning, Data Mining, Pattern Recognition and Text Learning paradigms. FSS formulates as follows: *Given a set of candidate features, select the best subset under some learning algorithm*. As the learning algorithm we are going to use na□ve Bayes [Duda & Hart, 1973] [Hand & Yu, 2001]. A good review of FSS algorithm can be found in [Liu & Motoda, 1998]. To test the FSS problem we will use the *chess* dataset from the UCI repository [Murphy & Aha, 1995], which has a total of 36 features and 699 instances.

Results: 4-Bit Fully Deceptive Function

Figure 2 shows the results for the 4-bit fully deceptive function using a population of 1000 individuals. Results for other size of populations are really similar.



(a) Constant Participation Functions (b) Other Participation Functions

Fig. 2. Results for the 4-bit Fully deceptive function

The best results are obtained with EDAs, while the worst are obtained with GAs. Using EDAs the maximum is reached in approximately 23 generations. On the other hand, using GAs, after 91 generations the maximum is never found. About our new hybrid approaches, we always reach the maximum, being the best Participation Function the dynamic one, which reaches the maximum in 27 generations and the worst Participation

[3] (http://www.satlib.org/benchm.html). *4blocksb.cnf* contain 24758 clauses, 410 propositional variables and is satisfiable.

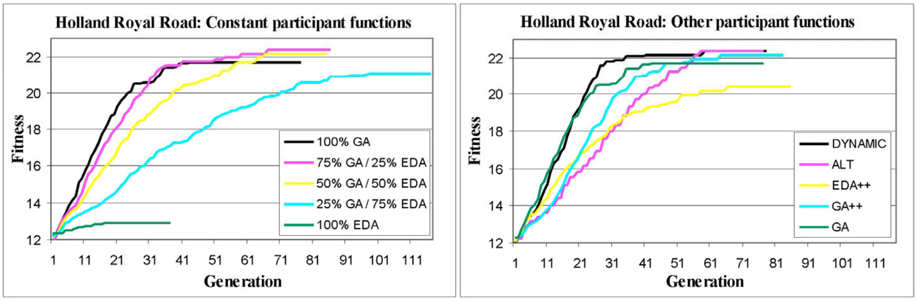
Function the constant ration with 25% EDA / 75% GA which reaches the maximum in 36 generations.

In conclusion, we always reach the maximum with our hybrid approaches and the bad results obtained with GAs only affect our hybrid in the number of generations required.

Results: 240 Bit Holland Royal Road

This problem is just the opposite of the previous one. As it is possible to see in Figure 3, with a population of 1000 individuals, the performance of GAs is much better than the performance of EDAs. With EDAs is only possible to achieve a fitness value of 12.91, while with GAs this value is 21.07. However, most of the hybrid approaches are better than GAs, being the best obtained value 22.37 with the DYNAMIC and the CONSTANT 75% GA / 25% EDA Participation Functions.

In conclusion, for the 240 bit Holland Royal Road problem our hybrid approach performs better than GAs and EDAs.

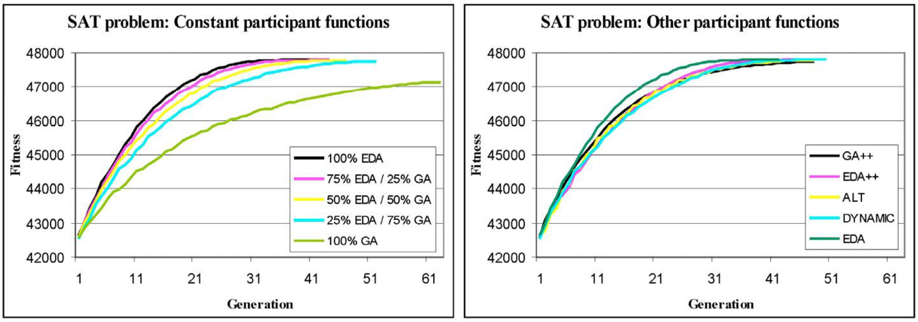


(a) Constant Participation Functions (b) Other Participation Functions

Fig. 3. Results for the Holland Royal Road problem

Results: SAT Problem

The experimental results obtained for the SAT problem are quite similar to the results of the 4-bit fully deceptive function (see Figure 4). The best results are obtained with EDAs, while the worst are obtained with GAs. Using EDAs the maximum (fitness = 47803) is reached in approximately 43 generations. On the other hand, using GAs, after 64 generations the maximum obtained is 47142.



(a) Constant Participation Functions (b) Other Participation Functions

Fig. 4. Results for the SAT problem

Our new hybrid approaches the best Participation Function is EDA++, which gives a fitness value of 48000. With the DYNAMIC, 25% EDA / 74% GA and 50% EDA / 50% GA Participation Functions the results are also good.

Feature Subset Selection

In the FSS problem GAs performance is better than EDAs performance. However, the hybrid solution using 50% EDA / 50% GA is better than both of them.

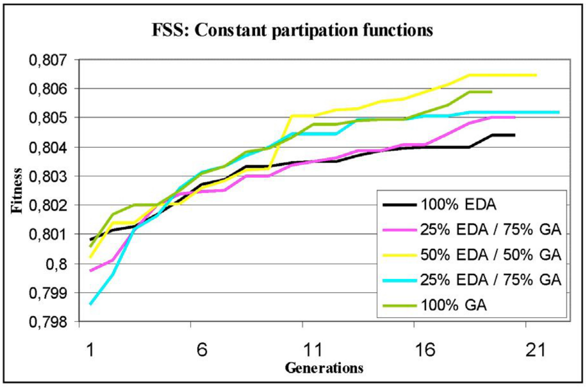


Fig. 5. Results for the FSS problem

4.2 Dynamic Participation Function: Evaluation

One of the most interesting aspects researched by this contribution is to know how the dynamic Participation Function performs for different kind of problems. This result provides an idea of how suitable is each of the methods for a specific kind of problem. And more useful, during the execution of the algorithm what is the performance base on the generation. Figure 6 shows the Percentage of GA participation in the Dynamic Participation Function for the four problems. In three of the four problems we can observe the same tendency, first we start to use the genetic algorithms, and after some generations the use of EDAs increase. This tendency is bigger in the problems in which GAs performs better than EDAs. However, it is necessary to remark that in the last generations, EDA algorithm always increases.

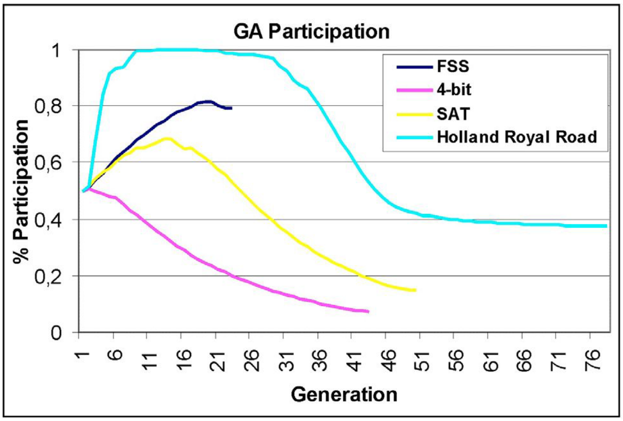


Fig. 6. Percentage of GA participation in the Dynamic PF

5 Conclusions and Future Work

In this paper we have proposed a new hybrid algorithm based on genetic and estimation of distribution algorithms. This new algorithm has been tested on four different problems: 4-bit fully deceptive function, Holland Royal Road, SAT problem and Feature Subset Selection. Although the hybrid algorithm proposed is composed by the simplest versions of both GA and EDA components and only works with bit-string individuals, the experimentation shows it is really promising and competitive. In most of the experiments we reach to the best of the values found by GAs or EDAs or even we improve them.

There is still a lot of further future work: Extend the implementation to support more sophisticated individual representations, for example with continuous genes, make new Participation Functions, make experimentation in more problems, implement a parallel version or use more complex GAs and EDAs in the hybrid solution.

Acknowledgements. The authors would like to acknowledge Raquel Hern□ndez for her work programming and testing the first stages of this algorithm, as well as her useful comments.

References

[Choe & Choi & Moon, 2003] Choe, H. and Choi, S. and Moon, B. (2003). A Hybrid Genetic Algorithm for the Hexagonal Tortoise Problem, in GECCO 2003, Genetic and Evolutionary Computation Conference, Springer Verlag, 2003, number 2723 in Lecture Notes in Computer Science, pp. 850 - 861.

[Deb & Golberg, 1993] Deb, K., & Goldberg, D.E. (1993). Analyzing deception in trap functions. In Withley, L.D. (Eds.), *Foundation of Genetic Algorithms 2* (pp. 93-108). San Mateo, CA: Morgan Kaufmann.

[Duda & Hart, 1973] Duda, R. and Hart, P. (1973). Pattern Classification and Scene Analysis. *JohnWiley and Sons*, 1973.

[Goldberg, 1989] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.

[Hand & Yu, 2001] Hand, D.J. and Yu, K. (2001). Idiot□s Bayes - not so stupid after all? *International Statistical Review*, 69(3):385□398, 2001.

[Hao & Lardeux & Saubion, 2002] Hao, J. and Lardeus, F. and Saubion, F. (2002). A Hybrid Genetic Algorithm for the Satisfiability Problem. 1rst International Workshop on Heuristics, Beinijg, 2002

[Holland, 1993] Holland, J. H. (1993). Royal Road functions. Internet Genetic Algorithms Digest, vol. 7, issue 22.

[Larraa□ga & Lozano, 2001] Larra–aga, P. and Lozano, J.A. (2001). Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. *Kluwer Academic Publisher.*

[Lim & Rodrigues & Xiao, 2003] Lim, A. and Rodrigues, B. and Xio, F. (2003). Integrated Genetic Algorithm with Hill Climbing for Bandwidth Minimization Problem, in GECCO 2003, Genetic and Evolutionay Computation Conference, Springer Verlag, 2003, number 2724 in Lecture Notes in Computer Science, pp. 1594□1595.

[Liu & Motoda, 1998] Liu, H. and Motoda, H. (1998). Feature Selection for Knowledge Discovery and Data Mining. *Kluwer Academic Publisher*.

[Mitchell et al., 1992] Mitchell, M., Forrest, S. and Holland, J.H. (1992). The royal road for genetic algorithms: Fitness Landscapes and GA performance. In *Proceedings of the First European Conference on Artificial Life*. Cambridge, MA: MIT Press/Bradford Books, 1992.

[Mh□lenbein, 1998] Mh□lenbein, H. (1998). The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303□346, 1998.

[Murphy & Aha, 1995] P. M. Murphy, P.M. and Aha, D.W. (1995). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/

[Rodriguez-Tello & Torres-Jimenez, 2003] Rodriguez-Tello, E. and Torres-Jimenez, J. (2003) ERA: An algorithm for reducing the epistasis of SAT problems, in GECCO 2003, Genetic and Evolutionary Computation Conference, Springer Verlag, 2003, number 2724 in Lecture Notes in Computer Science, pp. 1283□1294.

[Whitley & Starkweather, 1990] Withley, D. and Starkweather, T. (1990). GENITOR II: a distributed genetic algorithm, *Journal of Experimental and Theoretical Artificial Intelligence*, 2, 189-214.

[Zhang & Sun & Tsang & Ford, 2003] Zhang, Q. and Sun, J. and Tsang, E. and Ford, J. (2003). Combination of Guided Local Search and Estimation of Distribution Algorithm for Quadratic Assignment Problems, in GECCO 2003: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, pp, 42 □48.

Handwritten Numeral Recognition Based on Simplified Feature Extraction, Structural Classification, and Fuzzy Memberships

Chichang Jou and Hung-Chang Lee

Department of Information Management, Tamkang University, 151 Ying-Chuan Road,
Tamsui, Taipei County, Taiwan 251, Republic of China
{cjou,hclee}@mail.im.tku.edu.tw
http://www.im.tku.edu.tw

Abstract. Structural classification recognizes handwritten numerals by extracting geometric primitives that characterize each image. We propose a handwritten numeral recognition system based on simplified feature extraction, structural classification and fuzzy memberships, with the intention to find a small set of primitives without sacrificing the recognition rate. For each image, we first perform simplified preprocessing of smoothing and thinning to obtain a skeleton. For each skeleton, the following feature points are detected: terminal, intersection, and directional. We then extract the following primitives for each skeleton: loop, horizontal, vertical, leftward curve, and rightward curve. A fuzzy S-function is used as the membership function to estimate the likelihood of these primitives being close to the vertical boundary of the image. A tree-like classifier based on the extracted feature points, primitives and fuzzy memberships is then applied to recognize the numerals. Handwritten numerals in NIST Special Database 19 are recognized with correct rate between 87.33% and 88.72%.

1 Introduction

With numerous potential commercial applications, handwritten character recognition has been an active research field. With miscellaneous cultural backgrounds and extensive varieties of individual writing styles, the same character could be written in many forms and outlines. In addition, it is rather difficult for anyone to write the same character several times without any change. Suen et al. [12] observed that when ordinary people read an incomplete handwritten article, even after training, their error rate of recognition with regard to article contents is still about 4%. These all demonstrate the difficulty in handwritten character recognition. In this paper, we narrow down the problem to handwritten numeral recognition.

The handwritten numeral recognition problem has been studied from fuzzy logic [7], neural networks [2], rough set [6], statistics [15], structural classification [4], etc. Recently, there is a trend of combining two or more recognition methods to obtain a better recognition rate [1,2,13,14]. Feature extraction is the base technology of the above methodologies.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 372-381, 2004.

Siy and Chen [11] decomposed images of handwritten numerals into a set of fifteen branching features, which are of the following three categories: (1) straight lines: horizontal, vertical, positive slope, and negative slope, (2) circles: plain circle and circle on the left, on the right, above and below, and (3) open arcs: C-like, D-like, A-like, V-like, S-like, and Z-like. Their decomposition was based on the detection of the following feature points: tips, corners, and junctions. To increase recognition rate, Malaviya and Peters [7] increased the number of primitives of the above categories, and used multi-stage feature aggregation to describe each image using fuzzy rules. The increase of the number of primitives makes the computation more complex.

Hu and Yan [4] collected for each primitive local information of curvature, moving direction, length, etc. to form a primitive code of 11 elements. A global code of 5 elements was then deduced from the primitive codes to reflect global topological information, like the number of primitives. Based on the global codes, matching rules are designed for 103 prototypes and 26 subclasses. A neural network was employed to overcome the time-consuming design process. Mayora-Ibarra and Curatelli [8] divided each image into 16 equal-sized partitions, and then extracted 7 features from each partition to form a global vector of 112 features, which were then sent to a neural network for classification. Since some features are inter-dependent, they erased several features to obtain better classification results. Zhang and Chen [14] collected for each image two types of features: (1) global: middle line, concave, width, end points, branch points, and cross points; (2) local: tiny horizontal or vertical segments along character outer profiles.

We propose a handwritten numeral recognition methodology that combines simplified feature extraction, structural classification, and fuzzy memberships. We first integrate the smoothing algorithm of Hu et. al. [5] and the thinning algorithm of Datta and Parul [3] to obtain a skeleton for the bitmap image of each numeral. Simplified detection of feature points proposed by Hu and Yan [4] are then applied to separate a skeleton into paths. Following Nishiba and Mori [9], for each skeleton, through checking continuously whether a path and its concatenated path could be merged into one, we obtain a set of primitives for each handwritten numeral. Each primitive is then classified into one of the following five primitive types: loop, horizontal, vertical, leftward curve, and rightward curve. A fuzzy *S*-function is used as the membership function to estimate their likelihood of being close to the vertical boundary of the image. We propose a tree-like classification based on the primitives and the relative position of these feature points and primitives, through the membership function. The system is then applied to classify handwritten numerals from NIST Special Database 19. Our result is compared with those of other researches with the same dataset.

2 Smoothing and Thinning

To overcome unavoidable spurs and holes in digitized images, we adopt Hu et al.'s algorithm [4] to smooth image boundaries and to compensate their stroke width. It first eliminates the short and extra spurs and fills in small holes. A stroke width compensation algorithm is then applied to ensure that the width of each horizontal or vertical stroke is larger than three pixels. If a horizontal or vertical black stroke is less than four pixels, then the two points adjacent to it are the candidate points to be filled. If one candidate point has more neighboring black pixels than the other, then this point is filled.

Thinning will save the amount of memory space in storing the image, and also simplify later processing. Image thinning should preserve connectivity and should not shorten skeletal legs. We adopt Datta and Parul's thinning algorithm [3] to transform the smoothed bitmap image of handwritten numerals into a skeleton. This algorithm ensures one-pixel wide result, and preserves the connectivity. Let a pixel *P* and its eight neighboring points be represented as Figure 1(e).

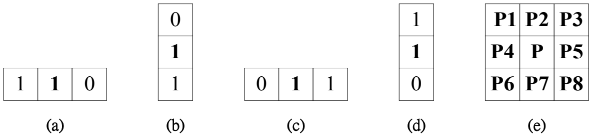


Fig. 1. Templates for Thinning Algorithm

For templates (a) to (d) in Figure 1, we say *P matches* a template if *P* is in the middle of that template. By checking neighboring pixels, when a point *P* satisfies any of the following conditions, *P*'s elimination will destroy the original connectivity of the image:

1. *P* matches one of Figure 1(a) to (d) and ((p2=0 and p3=1) or (p7=0 and p8=1))
2. *P* matches one of Figure 1(a) to (d) and ((p4=0 and p1=1) or (p5=0 and p3=1))
3. *P* matches one of Figure 1(a) to (d) and ((p2=0 and p6=1) or (p7=0 and p6=1))
4. *P* matches one of Figure 1(a) to (d) and ((p4=0 and p6=1) or (p5=0 and p8=1))

If *P* is not an end point and *P* does not satisfy any of the four conditions, then *P* is said to be *removable*. The thinning algorithm iteratively eliminates removable points until no more points could be eliminated. After smoothing and thinning, we obtain a one-pixel wide skeleton for each image.

3 Skeleton Decomposition

In Section 3.1, we decompose images into several simple paths and then reconstruct these paths for later processing. In Section 3.2, some of these paths are then merged. In Section 3.3, the resulting paths are then classified into one of the five primitive types.

3.1 Detection of Feature Points

We simplify Hu and Yan's algorithm [4] to obtain a set of feature points, which will be used for skeleton decomposition in the next subsection. The definitions of these feature points are as follows:

Definition: *T* (terminal) points are the points with exactly one black neighbor among their eight neighbors. *I* (intersection) points are the points with more than two black points among their eight neighbors. A *path* is a sequence of continuous black points, where both its starting point and ending point are either a T or I point. *D* (directional) points are those points in a path that change directions in either x-axis or y-axis.

Since the number of our intended resulting primitive types is small, we no longer detect the bending points, where the curvature of its belonging path exceeds a certain value, which were needed in Hu and Yan's algorithm. Starting from the left upper corner of the skeleton, from left to right and then from top to bottom, we calculate for all the black points the number of its neighboring black points to determine the set of all T and I points. Then for each T point, we find out all the D points along its path until another T or I point is encountered. Similarly, for each I point, we find out all the D points along its path until another T or I point is encountered. The skeleton is then divided by these feature points into several paths. Figure 2 shows the steps in the detection of these feature points. Figure 3 demonstrates the feature points and resulting paths for a skeleton of a handwritten image of '2'.

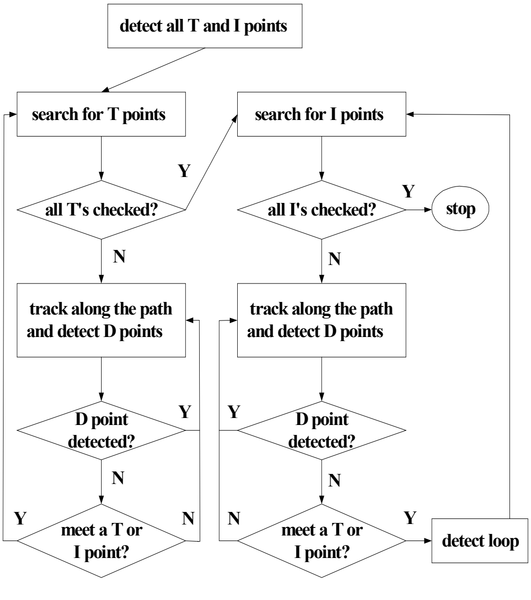


Fig. 2. Detection of Feature Points

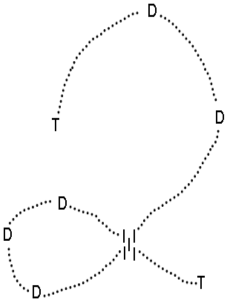


Fig. 3. Feature Points of a Handwritten “2”

$$\Delta(a,b) = sign\begin{vmatrix} x_a - x_p & x_b - x_p \\ y_a - y_p & y_b - y_p \end{vmatrix}$$

$$sign\ (x) \quad = \quad \begin{cases} -1 & if\ x < 0 \\ 0 & if\ x = 0 \\ 1 & if\ x > 0 \end{cases}$$

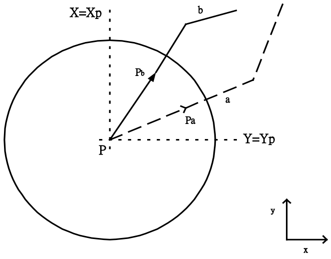


Fig. 4. Right-hand Rule of Path Merging

3.2 Merging of the Paths

We follow the right-hand rule proposed by Nishida and Mori [9] to decide whether two connecting paths of a skeleton could be merged. Figure 4 demonstrates how to use the rule as follows: Suppose *a* and *b* are two connecting paths, and *P* is the intersection point of *a* and *b*. Let P_a and P_b be points sufficiently close to *P* such that P_a and P_b are contained only in *a* and *b*, respectively. Let (x_p, y_p), (x_a, y_a), and (x_b, y_b) be the coordinates of *P*, P_a, and P_b. If $\Delta(a, b) = 1$, then these two paths could be merged as one, using *a*’s starting point as its starting point, and *b*’s ending point as its ending point. In other words, we always turn to the right at any joint of paths when we traverse a curve composed of concatenating paths. Note that unlike Nishida and Mori, we do not need to differentiate the convex of the curve as downward, leftward, upward, or rightward.

The merging algorithm works as follows: For each skeleton, we number all its paths from left to right and then from top to bottom. We test from the first path of the skeleton continuously whether the current path and its concatenated path could be merged according to the right-hand rule. Those checked paths are marked during this process. Then we reverse the process from the last path of the skeleton to check those unmarked paths. According to this merging algorithm, the skeleton in Figure 3 could be decomposed into primitives similar to the primitive D, OL, and H in Figure 5.

3.3 Primitive Classification

After the detection of feature points and merging of paths, each skeleton is decomposed into several primitives. Siy and Chen [11] proposed the set of fifteen sufficient branch primitives in Figure 5 for handwritten numeral recognitions. We instead propose a set of five primitive types for the classification.

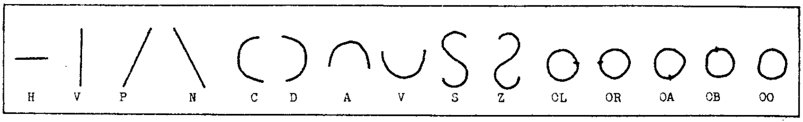


Fig. 5. Sufficient Primitives for Handwritten Numeral

If the starting point and ending point of a primitive are the same, then this primitive is classified a 'loop'. For the rest primitives, we first classify them into lines or curves as follows. Let the distance *d(P,L')* between a point *P* and a straight line *L'* be defined as the distances *d(P,P')* between *P* and their vertical intersection point *P'*. We define the distance *d(L,L')* between a primitive *L* and a straight line *L'* as the maximal of the distances *d(P,L')* for all points *P* in *L*. For each primitive *L*, we define a line L_c as the closest straight line of *L* if L_c has the minimum value of *d(L,L')* among all *L'*. L_c could be easily obtained by applying the least squares method [10]. Suppose $d(L,L_c)$ is ε_1, and the distance between the two intersecting points of *L* and *L'* is ε_2. Let R_L be the ratio of ε_1 and ε_2. If R_L is greater than a threshold R_T, meaning *L*'s vertical variation over horizontal variation is large, then we classify it as a curve. Otherwise, we classify it as a line.

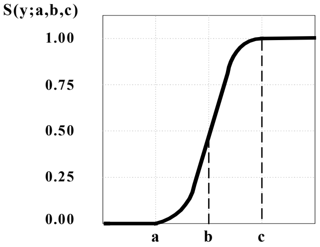
We classify a line into vertical or horizontal by checking the angle between the line and the x-axis. If the angle is less than a threshold θ_T, then we classify the line as horizontal. Otherwise, it is classified as vertical. For curves, we classify them into leftward or rightward curves by checking the x-coordinates of its starting and ending points. If the starting and ending points are both in the right hand side of the D-point of the curve, then it is classified as a leftward curve. Otherwise, it is classified as a rightward curve.

4 Tree-Like Classification

Before our investigation into the fuzzy relative closeness to the vertical boundary for feature points and primitives, we normalize the image length and width to be within [*0,1*]. Let the x-coordinates and y-coordinates of all points in a skeleton be bounded by X_{min}, X_{max}, and by Y_{min}, Y_{max}. The coordinates (*X, Y*) of a point would then be normalized by the following formulae:

$$X_n = (X - X_{\min})/(X_{\max} - X_{\min}) \qquad Y_n = (Y - Y_{\min})/(Y_{\max} - Y_{\min})$$

Using the above normalized coordinates of a point *P*, we then train one *S*-function, like the one in Figure 6, as the fuzzy membership function to determine the likelihood of *P* being close to the top of the skeleton. In the *S*-function, *b* is always set to be *(a+c)/2*. If *a=0.3*, and *c=0.9*, then for a point with y_n = *0.7*, it is *0.78* close to the top of the image. We estimate how close a primitive is to the top of the image by averaging the memberships for its starting and ending points. When the membership value is greater than or equal to 0.5, we say the primitive is *close to the top*. Otherwise, we say it is *close to the bottom*.



$$S(y;a,b,c) = \begin{cases} 0 & 0 <= y <= a \\ 2\left(\dfrac{y-a}{c-a}\right)^2 & a < y <= b \\ 1-2\left(\dfrac{y-c}{c-a}\right)^2 & b < y <= c \\ 1 & c <= y \end{cases}$$

Fig. 6. Fuzzy Membership Function

We first eliminate primitives useless for the classification, like those paths with length less than a threshold ratio *r* of *(Y_{max} – Y_{min})*. A tree-like classifier in Figure 7 is then constructed through identifying the first one or two primitives of the skeleton to classify the images, where '*o*' represents the loop, '−' the horizontal, ' | ' the vertical, '(' the leftward curve, ')' the rightward curve, and '*ϕ*' no primitives found. We scan the skeleton from left to right, and then from top to bottom, to find the first T-point. If a T-point is found, then its associated primitive must belong to one of the following four: '(', ')', ' | ', and '−'. Otherwise, we assign loop '*o*' as its associated primitive. According to the first primitive, we classify the image into the first layer nodes. For first layer nodes with more than three digits, we classify them according to the next connecting primitive to obtain the second layer nodes.

For end nodes with one digit, the classification finishes. For end nodes with two or three digits, further classification is needed. Let *node(d_1, d_2)* denote the node with digits d_1 and d_2, and *node(d_1,d_2,d_3)* the node with digits d_1, d_2, and d_3. For *node(8,0)*, the rule is:

node(8,0): If we could find more than one 'o' primitive, the image is classified as '8'. Otherwise, it is classified as '0'.

For the other end nodes, the following rules are then constructed based on the membership for the relative position of the feature points or primitives:

node(7,0): If the ending point of ')' is close to the bottom, then it is classified as '7'. Otherwise, it is classified as '0'.

node(3,6): If the starting point of the second ')' is close to the bottom, then it is classified as '6'. Otherwise, it is classified as '3'.

node(2,7): If '−' is close to the bottom, then it is classified as '7'. Otherwise, it is classified as '2'.

node(3,4): If the ending point of ')' is close to the top, then it is classified as '3'. Otherwise, it is classified as '4'.

node(6,0): If '(' is close to the top, then it is classified as '0'. Otherwise, it is classified as '6'.

node(2,6,8): If the middle point of primitive ‘o’ is close to the top, then it is classified as ‘2’. Otherwise, if the I-point is close to the top, then it is classified as ‘8’. Otherwise, it is classified as ‘6’.

node(4,9): If the starting point of ‘(’ is close to the bottom, then it is classified as ‘4’. Otherwise, it is classified as ‘9’.

node(6,9): If the middle point of ‘o’ is close to the top, then it is classified as ‘9’. Otherwise, it is classified as ‘6’.

node(4,5): If the starting point of ‘)’ is close to the top, then it is classified as ‘4’. Otherwise, it is classified as ‘5’.

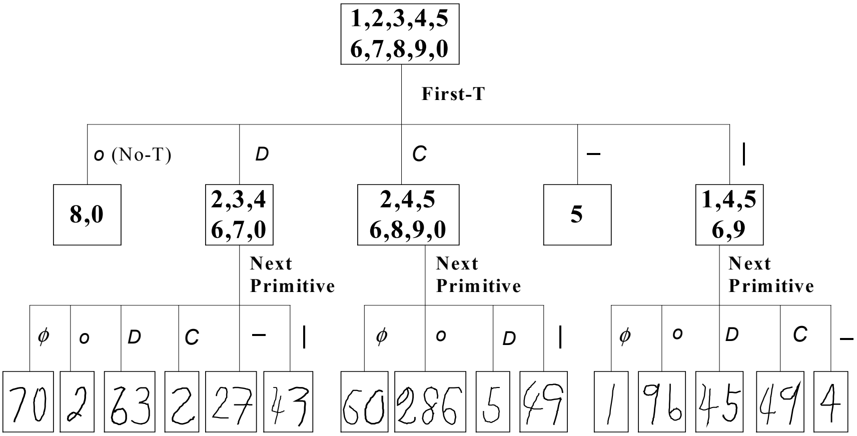


Fig. 7. The Tree-like Classifier

5 Experimental Results

The handwritten numerals we use to test our system are from NIST (National Institute of Standards and Technology) Special Database 19[1], which collected 810,000 handwritten characters of 3,600 people enclosed in forms. Each character was scanned in 300 dpi resolutions to obtain a 128*128 bitmap image. The number of handwritten numerals in it is 60,089.

We use 2,000 images as our training set to obtain the values for the thresholds R_T and θ_T, the ratio r of the useless short primitives, and the parameters a, c of the S-function with the best classification result, through trial and error. The parameters are then applied to classify the whole database. We compute the following performance parameters: (1) Correct rate: the ratio of correctly classified numerals. (2) Error rate: the ratio of wrongly classified numerals. (3) Rejected rate: the ratio of unclassifiable numerals. (4) Reliability: the ratio of correct rate and the sum of correct rate and error rate. Ten different training set are used to train the classification. Table 1

[1] http://www.nist.gov/srd/nistsd19.htm

demonstrates our best and worst results, compared with the results of Hu et al. [4], Mayora-Ibarra et al. [8], and Zhang et al. [14]. Note that Hu et al. and Zhang et al. used NIST Special Database 3, a discontinued subset of NIST Special Database 19, with only 20,852 images of handwritten numerals.

Even though the number of primitive types in our system is much less than those of other researches, our correct rate is only a little lower than the other three cases. The reliability of our result is more stable than their results. The fuzzy closeness of feature points and primitives to the vertical boundary from our membership function helps in the final classification for the choice among two or three digits. Because the number of final primitive types is small, our rejected rate is low. That caused our error rate higher than the worst case of Hu et al., which is accompanied with a higher rejected rate.

Table 1. Comparison of Recognition Results

	Hu (best)	Hu (worst)	Mayora-Ibarra (best)	Mayora-Ibarra (worst)	Zhang	Jou (best)	Jou (worst)
Correct rate	97.29%	88.79%	96.6%	89.6%	98.10%	88.72%	87.33%
Error rate	2.71%	0.25%	3.4%	10.4%	0.75%	9.36%	10.65%
Rejected rate	0.00%	10.96%	0.0%	0.0%	1.15%	1.92%	2.02%
Reliability	97.29%	99.72%	96.6%	89.6%	99.24%	90.46%	89.13%

6 Conclusions

We designed and implemented a handwritten numeral recognition system based on simplified feature extraction, structural classification and fuzzy memberships. We integrated the smoothing algorithm of Hu et al. and the thinning algorithm of Datta et al. to obtain a skeleton for each image. Hu and Yan's algorithm was then simplified to catch feature points of each skeleton and to decompose it into several paths. Simplified right-hand rule is then applied to merge these paths. We classified the resulting paths into a set of five primitive types. As far as we know, this number of feature primitives is the smallest up to now. Since the number of our primitive types is much smaller, the classification is also simpler. A fuzzy *S*-function was then applied to these primitives to estimate the likelihood of a point or primitive being close to the top of the image. Finally, a tree-like classifier is utilized to classify these numerals based on the primitives and memberships. Handwritten numerals in NIST Special Database 19 are recognized by our system with 88.72% correct rate. As the system integrates several modules, we do not have a detailed efficiency comparison with other systems. However, with simplified detection of feature points and convex differentiation, our system is comparatively efficient.

The following are some future research ideas: Our trial-and-error training method to obtain the parameters of the *S*-function and the thresholds (r, R_T and θ_T) takes time. A more automatic and systematic method is needed. The fuzzy membership for the horizontal boundary could be incorporated to increase the correct rate. The extended studies of recognizing digital strings, overlapping digits and digits inside form lines would be very useful in commercial applications. Another interesting topic is to study the influence of cultural background on the features of handwritten characters

References

1. Cao, J., Ahmadi, M., Shridhar, M.: Recognition of Handwritten Numerals with Multiple Feature and Multistage Classifier. Pattern Recognition, Vol. 28, (1995) 153-160
2. Chen, G.Y., Bui, T.D., Krzyzak, A.: Contour-based Handwritten Numeral Recognition Using Multiwavelets and Neural Networks. Pattern Recognition, Vol. 36, (2003) 1597-1604
3. Datta, A., Parul, S.K.: A Robust Parallel Thinning Algorithm for Binary Images. Pattern Recognition, Vol. 27, (1994) 1181-1192
4. Hu, J., Yan, H.: Structural Primitive Extraction and Coding for Handwritten Numeral Recognition. Pattern Recognition, Vol. 31, (1998) 493-509
5. Hu, J., Yu, D., Yan, H.: Algorithm for Stroke Width Compensation of Handwritten Characters. Electronics Letters, Vol. 32, (1996) 2221-2222
6. Kim, D., Bang, S.-Y.: Handwritten Numeral Character Classification Using Tolerant Rough Set. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, (2000) 923-937
7. Malaviya, A., Peters, L.: Fuzzy Feature Description of Handwriting Patterns. Pattern Recognition, Vol. 30, (1997) 1591-1604
8. Mayora-Ibarra, O., Curatelli, F.: Handwritten Digit Recognition by Means of a Holographic Associative Memory. Expert Systems with Applications, Vol. 15, (1998) 399-403
9. Nishida, H., Mori, S.: Algebraic Description of Curve Structure. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, (1992) 516-533
10. Press, W.H., Teukolosky, S.A., Vetterling, W.T., Flannery, B.P.: Numeric Recipes in C: The Art of Scientific Computing. Cambridge, Cambridge University Press (1996)
11. Siy, P., Chen, C.S.: Fuzzy Logic for Handwritten Numeral Character Recognition. IEEE Trans. on Systems, Man and Cybernetics, (1974) 520-574
12. Suen, C.Y., Shinghal, R., Kwan, C.C.: Dispersion Factor: A Quantitative Measurement of the Quality of Handprinted Characters. Proceedings of International Conference on Cybernetics and Society, (1977) 681-685
13. Wang, J., Yan, H.: A Hybrid Method for Unconstrained Handwritten Numeral Recognition by Combining Structural and Neural 'Gas' Classifiers. Pattern Recognition Letters, Vol. 21, (2000) 625—635
14. Zhang, P., Chen, L.: A Novel Feature Extraction Method and Hybrid Tree Classification for Handwritten Numeral Recognition. Pattern Recognition Letters, Vol. 23, (2002) 45-56
15. Zhang, R., Ding, X.: Offline Handwritten Numerical Recognition Using Orthogonal Gaussian Mixture Model. IEEE International Conference on Image Processing, Vol. 1, (2001) 1126-1129

Using Chaos Theory for the Genetic Learning of Fuzzy Controllers

Alfons Schuster

University of Ulster, Faculty of Engineering, School of Computing and Mathematics
Shore Road, Newtownabbey, Co., Antrim BT37 0QB
Northern Ireland
a.schuster@ulster.ac.uk

Abstract. This paper describes a study we have undertaken in the field of intelligent, hybrid systems. The system presented, which is basically a control system, applies ideas from the fields of fuzzy logic, genetic algorithms, and chaos theory. The system is fully implemented using the Delphi programming language. First results produced by the system are encouraging in our view.

1 Introduction

The work presented here relates to a larger research effort that aims for the design and implementation of system control software in the field of robotics. The system presented here is regarded as a software simulation presenting an intermediate step, or groundwork, towards a practical robotic implementation in the future. We therefore from now on refer to the system as *RobSim*. Our general research strategy follows an incremental and iterative approach in which we continuously increase the level of complexity, learning from the problems emerging at the different levels. For example, the current effort is based on the experience we gained from various experimental studies we have undertaken in the past [1, 2, 3]. Very basically, *RobSim* can be viewed as an integrated, intelligent, hybrid system that uses various soft computing techniques in a control environment. This hybrid system is composed around three basic components, a genetic algorithm component, a non-linear dynamic component, and a fuzzy control system (FCS) component.

Since the FCS component is central to the system, we start with some observations related to this component. The design of FCSs can be problematic for various reasons. Problems may be due to the complexity of the domain, the accessibility and availability of domain knowledge or domain experts, the number of rules needed for a rule base, or the consistency and maintenance of such a base, for example. Rules are not the only means by which knowledge is captured in a FCS. Fuzzy sets, their shape and arrangement, as well as the inference mechanisms used in a system are also important [4]. FCS design also very often has a strong trial and error nature in which system designers often play a vital role. A consequence of this fact is that it is very often possible to generate multiple FCS solutions for the same problem. For example, the only difference between two FCS solutions could be the defuzzification technique

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 382-391, 2004.

they employ, but it also could be the slightly different shape of particular fuzzy sets. Another observation is that many FCSs show similarities in their dynamic behaviour. For example, the dynamic behaviour illustrated in Fig.1 could be from a FCS controlling the movement of a robot arm trying to grasp an object on an assembly line, but also from a system controlling the temperature in a room. This illustration actually is taken from an example application provided with the commercial fuzzy logic tool CubiCalc 2.0 that was used in this study. Note however that the circled line in the figure has been added manually to ease forthcoming discussions.

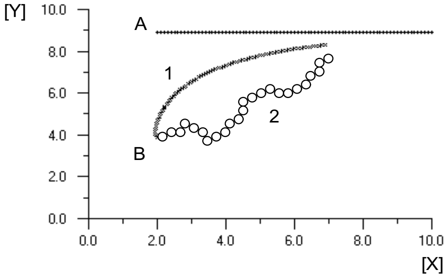


Fig. 1. Dynamic behaviour of an example FCS that ships with the commercial fuzzy logic tool CubiCalc 2.0.

Let Fig.1 illustrate the trajectories of two objects, *A* and *B*, moving from left to right in time. *Y* and *X* in the figure define a co-ordinate system. The objective of object *B* is to approach and finally catch object *A*. Both objects move with constant, but individual speeds, and so a dot or circle at position (*X*, *Y*) in the figure represents the position of an object in time. For simplicity object *A* moves on a straight line. Object *B* has to be more flexible due to the definition of its task. Fig.1 illustrates two trajectories for object *B*. Fig.1 indicates that following either trajectory object *B* finally approaches object *A*, and so both trajectories provide a solution to the given task. The solution finally selected however could be the FCS that produces trajectory 1, because for many problems FCS designers prefer a system that converges towards a solution with some smoothness. Simply imagine the two trajectories as being proposed solutions for the robot arm mentioned before. It is not difficult to select the one more appropriate for the task. A synopsis of these observations could be: (1) It is very often possible to generate multiple FCS solutions for a problem, (2) FCSs applied to different problems may show similarities in their dynamic behaviour, and (3) convergence with a certain degree of smoothness can be a requirement in a FCS. These observations form the basis for this work, which, in broad terms, can be summarised as a proposal for the automation of the process of designing FCSs. The means by which we aim to achieve this goal are:

1. The convergence of a proposed FCS solution is examined by a measure of convergence. This measure shows resemblance to the so-called Lyapunov Exponent in chaos theory.
2. The smoothness of a solution is determined by a fractal dimension algorithm.
3. The paper presents an integrated system for the generation, evaluation, and selection of potential FCS solutions.

2 Measures from Chaos Theory

Chaos theory has its origins in the study of non-linear dynamical systems. It obtained increasing attention within the natural sciences about four decades ago. Through the fast progress in computer technology within this period it was possible to investigate more and more complex systems with increasing efficiency [5]. Lorenz, for example, investigated the extent to which weather is predictable [6]. Nowadays chaos theory is studied in many domains including medicine and logic, for example [7, 8]. Out of these studies emerged a variety of new concepts and measures.

2.1 A Measure of Convergence

In chaos theory the so-called Lyapunov Exponent λ is a useful measure to assist in the distinguishing between different types of trajectories of dynamic systems [9]. It is based on the mean exponential rate of divergence of two initially close trajectories and describes the dynamic of a system qualitatively as: $\lambda < 0$, the orbit is attracted to a stable fixed point or a stable periodic orbit; $\lambda = 0$, the orbit is a neutral fixed point. The system is in some sort of steady state mode, like a satellite in a stable orbit, for example; $\lambda > 0$, the orbit is unstable and chaotic. Nearby points, no matter how close, will diverge to any arbitrary separation. The measure used in this study bears similarities in its interpretation to the description given before. Let Fig.2 illustrate the trajectories of two arbitrary objects *A* and *B*, and let $d(t_0)$ be the distance *d* between objects *A* and *B* at time t_0, and $d(t_0+\Delta t)$ the distance at time $t=t_0+\Delta t$.

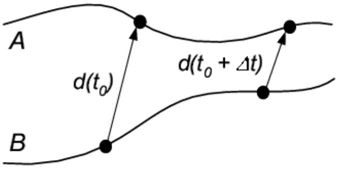


Fig. 2. The development of two trajectories, one from object *A* one from object *B*, over time.

To describe such a development of the distance between two objects over time this study uses a so-called measure of convergence (*MOC*):

$$MOC = \frac{1}{N-1}\sum_{n=1}^{N-1}\log_{10}\left|\frac{d_{n+1}}{d_n}\right| \tag{1}$$

With *N* being the number of data points in the time series. This *MOC* is not used to determine whether a system is chaotic or not. It is rather interpreted in the context of the similarities it bears with the Lyapunov Exponent mentioned earlier. The features of this measure that could be useful in this study are: *MOC* < 0, may be an indicator for a system that produces convergent trajectories. *MOC* = 0, might indicate a system that is in some sort of steady state mode, for example, objects *A* and *B* moving on two parallel lines. *MOC* > 0, very likely an indicator for a system that produces non-convergent trajectories [1].

2.2 A Measure of Smoothness

The study of so-called fractals may provide a possibility for quantifying the shape of a trajectory in terms of its smoothness, or jaggedness, respectively. Very generally, fractals are patterns or structures which, when being dealt with mathematically, produce results or properties that are difficult to be interpreted, or conflicting with predictions of traditional mathematics. Mandelbrot, for example, associates these pathological structures with forms that can be found in nature [10]. Hausdorff and Besicovitch on the other hand came forward with a general definition for the calculation of a (fractal) dimension for such objects. Their definition of a fractal dimension is based on an investigation of how geometric figures fill the space in which they are represented [9]. Nowadays there exist many definitions, and variations of them, for measurements of fractal dimensions. This paper, for instance, uses a method proposed by Gough [7]. The geometric objects investigated here are time series representing the development of the distance between the trajectories of two objects, for example the time series illustrated in Fig.3-a.

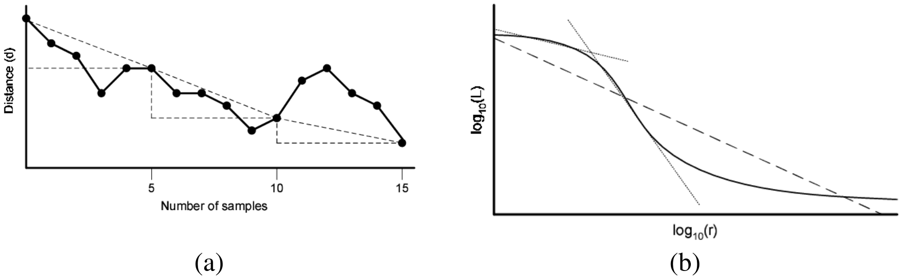


Fig. 3. (a) distance development of two trajectories represented by a time series, and length estimation using a ruler length of five, and (b) extraction of a fractal dimension.

Fig.3-a illustrates that individual distance measurements are connected to a continuous line. Gough's method calculates a fractal dimension from such a line. Initially the method determines different estimates of the length *Lr* of the line by measuring it with different so-called rulers of length *r*. The time series in Fig.3-a for instance is measured with a ruler of length five. In simple terms a single length estimate *Lr* is a summation of hypotenuses. In order to extract a fractal dimension from such a diagram the method plots the logarithm of the length estimates *log10Lr* against the logarithm of the ruler length *log10r*. This is illustrated in Fig.3-b. The establishment of a fractal dimension from such a diagram is not that simple. The traditional definition by Hausdorff and Besicovitch leads towards using the slope of a regression line (dashed line in Fig3-b) through the data points as an approximation for a fractal dimension. Other researchers came up with other interpretations. Kaye for example generates regression lines and fractal dimensions for separate regions in a plot (the two dotted lines in Fig.3-b for example), and compares these fractal dimensions with the features of "structure" and "texture" in fine-particle science [11]. This paper follows Kaye's view, and so it could be said that a measurement with longer rulers identifies the global behaviour (structure) of the distance between two trajectories. On the other hand, measurement with smaller rulers provides information about the behaviour of the distance function at smaller scales (texture). We investigated the *MOC* and the fractal dimension algorithm on a number of FCSs and other models for quality assessment in previous studies [1].

3 System Overview

It was mentioned earlier that *RobSim* is designed around three basic components, a genetic algorithm component, a FCS component, and a non-linear dynamic component. Fig.4 illustrates the assembly of these components into an integrated system.

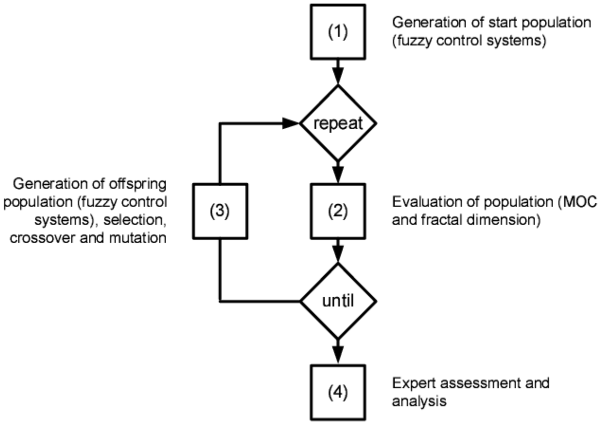

(1)
Generation of start population (fuzzy control systems)
repeat
Generation of offspring population (fuzzy control systems), selection, crossover and mutation
(3)
(2)
Evaluation of population (MOC and fractal dimension)
until
(4)
Expert assessment and analysis

Fig. 4. An integrated view of the *RobSim* system.

Fig.4 basically illustrates a genetic algorithm. Step (1) initially generates a start population with a pre-defined number of FCSs. This step uses prior knowledge, for example knowledge about input/output variables, but also employs a random element, for example when generating the shape of fuzzy sets representing a system variable. Step (2) and step (3) are included in a repeat-until loop with pre-defined STOP settings. Step (2) estimates the quality of each FCS in a population. This analysis is based on values for *MOC* and fractal dimensions. Step (3) uses typical genetic algorithm procedures to select and modify solutions that indicate as being better than other solutions in order to achieve further improvement. For the current system a system developer would evaluate the final proposal of the system in step (4). We already mentioned that FCS design often involves a lot of trial and error. The same can be said about genetic algorithm design where settings for crossover rate, number of iterations, etc. are similarly problematic. The results presented here are based on the following settings: number of iterations n =

35, number of FCSs in a population = 50, crossover rate = 20, mutation rate = 5, time series = 200 data points. The process in Fig.4 can be implemented at different levels of complexity, for example for rule base generation, weight assignment, or the shaping of fuzzy sets. The general strategy is to continuously increase the level of complexity, and to learn from the problems emerging at the different levels. The study presented here concentrates on the shaping of fuzzy sets. In order to understand this process we use an example application that comes with the commercial CubiCalc tool.

3.1 The Control Task

The example tool describes the control task as a scenario where a dog chases a cat by adjusting its azimuth in accordance with the relative direction of the cat. The rational of the system is simple: correct the dog's azimuth in direct proportion to a tracking error. Fig.1 actually did illustrate such a "dog chasing cat" scenario. In the tool the control scenario is not modelled in too much detail. For example, ideally the dog should actually catch the cat, but in the tool the dog stays behind the cat, "only" its path merges with the direction of the cat's path. Overall the control system for this task is not too complex. It contains one input variable called *TrackingError*, and one output variable called *AzimuthAdjust*. The rule base of the system contains five, relatively simple rules of the form: *If TrackingError is LargePositive then make AzimuthAdjust LargePositive*, for example. The input variable and the output variable are represented by fuzzy sets. Fig.5-a, and Fig.5-b illustrate these representations.

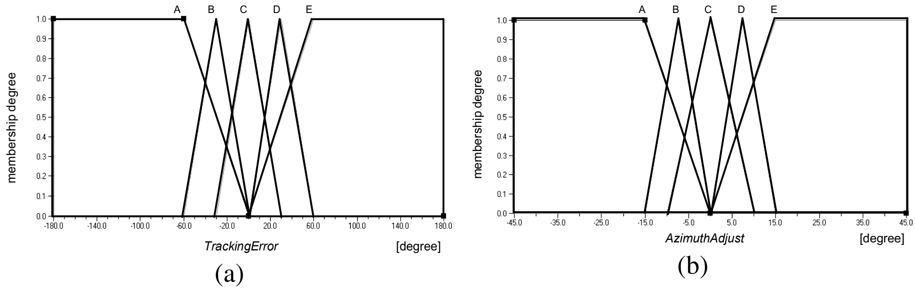
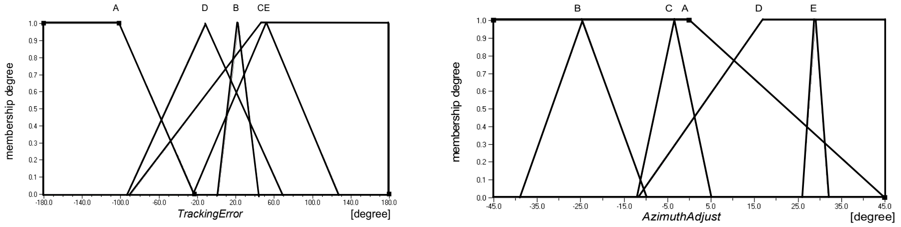


Fig. 5. Fuzzy sets for (a) the input variable *TrackingError*, and (b) the output variable *AzimuthAdjust* (taken from the commercial fuzzy logic tool CubiCalc 2.0).

Fig.5-a and Fig.5-b indicate that both variables are defined by five fuzzy sets. Every fuzzy set is defined by three points. Within the tool these fuzzy sets carry specific names. For example, the fuzzy sets for the variable *TrackingError* are labeled *LargeNegative*, *SmallNegative*, *NearZero*, *SmallPositive* and *LargePositive*. For simplicity forthcoming discussions refer to these fuzzy sets by mentioning the variable and one of the labels *A*, *B*, *C*, *D*, and *E*, where, from left to right, *A* stands for the fuzzy set to the very left and *E* for the fuzzy set to the very right. Note that this sequence is determined by the location of the middle point of a fuzzy set. *RobSim* determines and records such sequences for every FCS it generates.

4 Testing and Results

The strategy of experimenting with scenarios of increasing complexity was mentioned before, and so was the shaping of the fuzzy sets for the input and the output variable as a starting point. *RobSim* initially produces a start population containing a pre-defined number of FCSs (Fig.4). The shaping of the fuzzy sets for the input and output variable in the start population involves a random element that requires a bit of explanation. The "correct" order for the fuzzy sets for both variables, from left to right, is *A*, *B*, *C*, *D*, *E*. This sequence is determined by the location of the middle points of the fuzzy sets. The random process creates: (a) fuzzy sets of different shapes, and also (b) sequences in which the fuzzy sets may not be in the correct order. For example, Fig.6-a, and Fig.6-b are taken from a FCS in the start population.



(a)

(b)

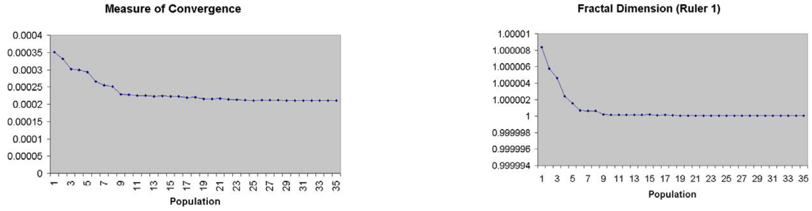
Fig. 6. Shape and sequence of (a) input fuzzy sets in start population (sequence: *A*, *D*, *B*, *C*, *E*), and (b) output fuzzy sets in start population (sequence: *B*, *C*, *A*, *D*, *E*).

Fig.6-a and Fig.6-b illustrate that the shapes of these fuzzy sets is quite different from the shape of the original fuzzy sets (Fig.5-a and Fig.5-b). But it is not only the shape of the fuzzy sets that is different. There is also a difference in the actual sequences. For example, the sequence for the fuzzy sets in Fig.6-a is *A*, *D*, *B*, *C*, *E*, and for those in Fig.6-b the sequence is *B*, *C*, *A*, *D*, *E*. The genetic algorithm continues from the start population through *n* iterations to the final population. Each iteration applies crossover, mutation and selection. The aim of this process is to generate FCSs of increasing problem-solving potential by shaping the fuzzy sets in the different populations towards the shape of the fuzzy sets from the original example application.

4.1 Genetic Algorithm Performance

This information can be extracted from the development of the *MOC* and the two fractal dimensions over the number of iterations. The values for these measures are expected to decrease with an increasing number of iterations (Section 2). In the system these values are actually added up for every FCS in a population, and the sums obtained are then averaged. Fig.7-a, and Fig.7-b illustrate the development of these measures over 35 iterations for a particular scenario. Section 2 mentioned that a fractal dimension depends on the scale (ruler) under which a time series is measured.

Fig7-b illustrates the development using one particular scale (Ruler 1). Measurement on other scales always produced a similar behaviour.



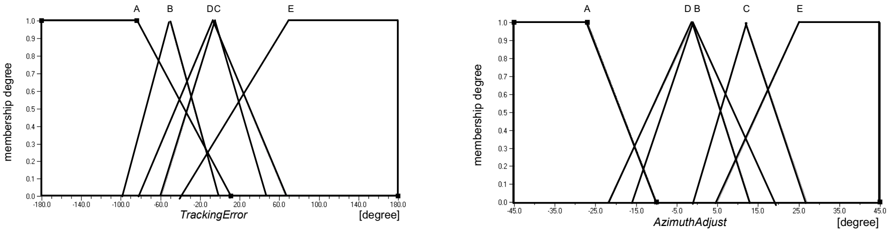
(a) (b)

Fig. 7. (a) development of the *MOC*, and (b) development of the fractal dimension measure in different populations over 35 iterations.

Fig.7-a, and Fig.-b clearly illustrate the expected behaviour. This is a positive result, as it supports our assumptions stated in Section 1 and Section 2.

4.2 Shaping and Sequence of Fuzzy Sets

Fig.8-a and Fig.8-b illustrate the shape of fuzzy sets generated by *RobSim* in the final population. Note that these figures stand representatively for the final population.



(a) (b)

Fig. 8. (a) input fuzzy sets in final population (sequence *A*, *B*, *D*, *C*, *E*), and (b) output fuzzy sets in final population (sequence *A*, *D*, *B*, *C*, *E*).

Fig.8-a and Fig8.-b provide further support for the proposed approach, because both figures illustrate a significantly higher degree of similarity with the fuzzy sets in the original system (Fig.5-a and Fig.5-b) than the fuzzy sets in the start population (Fig.6-a and Fig.6-b) do. Based on these observations it could be said that the *RobSim* approach “drives” the shaping of the fuzzy sets into the right direction. But what about the sequence of the fuzzy sets? Initially the outcome in this department is not so positive, because the fuzzy sets are not in the desired order. For example, in Fig.8-a the sequence is *A*, *B*, *D*, *C*, *E*, and in Fig.8-b it is *A*, *D*, *B*, *C*, *E*. This is a problem, but it can be solved, at least to a considerable extent. Although the proposed strategy is to go from more simple tasks to more challenging tasks the generation of the fuzzy sets is actually “over-complicated”. Chen and Hwang, for example, mention that there almost always is a logical order for the description of variables in fuzzy systems [12]. For example, in a temperature context the logical order would be: very cold, cold, warm, very warm, and **not**: cold, very warm, very cold, warm. Similarly, the logical description of the variables *TrackingError* and *AzimuthAdjust* is *A*, *B*, *C*, *D*, *E*, and nothing else. It is relatively unproblematic to program this ordering into *RobSim*. Why has this not be done in the first place? Because the "over-complication" has a purpose. One of our aims at this stage was to investigate whether the proposed approach works into the right direction? When experimenting with the commercial tool it turned out that just altering the shape of the fuzzy sets a bit, without changing their sequence, still produces systems of reasonable quality. Whether the proposed approach has the ability to drive the system to a solution was much better observable through a slightly increased level of complexity, namely by allowing un-ordered sequences.

4.3 Proposed Solutions

This final investigation examines the dynamic of FCSs generated by *RobSim*. Fig.9-a, and Fig.9-b stand representatively for a population. Individually they represent the dynamic of FCSs generated in the start population (Fig.9-a) and FCSs generated in the final population (Fig.9-b).

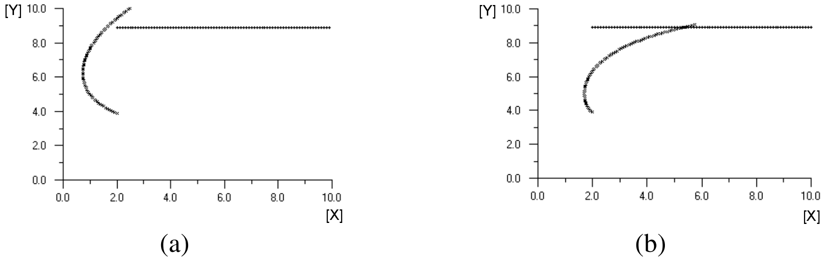


Fig. 9. Control scenario in (a) start population, and (b) final population. The start population is generated randomly by the system. The final population is generated in a guided process.

Fig.9-a and Fig.9-b support the proposed *RobSim* approach, because the dynamic behaviour of the FCS representing the final population (Fig.9-b) illustrates a significantly higher degree of similarity with the dynamic generated by the commercial tool (Fig.1) than the FCS representing the start population (Fig.9-a) does.

5 Related Work and Summary

Work that is relevant here comes from Chen & Hwang and Miller, for example [12, 13]. Chen and Hwang indicate that it is nearly always possible to describe FCS applications in completely different domains with a relatively small number of very often similar fuzzy sets. Miller supports this observation by identifying the number seven plus/minus two as a benchmark in many complex situations. For example, instead of lengthy explanations chess players often only mention a small number of key features of a game. Similarity and simplicity play a major role in the presented approach too. A good implementation of both aspects would keep the complexity of the system low, without loosing too much in terms of expressiveness and generality. Schuster presents further material about self-similarity in chaos theory and so-called adaptive fuzzy sets in the context of intelligent systems [2]. This paper argues that a set of fuzzy sets used for the description of a system variable often can be used for the same variable at different scales, but also very often for a different variable in a completely different domain. In simple terms self-similarity and adaptive fuzzy sets would increase the flexibility of the system, moving it from a single-domain system to a multiple-domain approach. For example, we currently apply the presented approach to a control task where the problem is to stably suspend a metal object in air midway between a magnet and the ground. This problem is simple in principle but difficult to solve since the system dynamics are highly nonlinear. Finally, in the field of FCSs (and also in robotics) researchers nearly always emphasise the trial and error nature of the development process and the importance of the system developer. The integration of the techniques presented in *RobSim* around a genetic algorithm seems to suit this problem quite well.

References

1. Schuster, A., Blackburn, W.T., Seguí Prieto, M. A Study on Fractal Dimensions and Convergence in Fuzzy Control Systems. Journal of Telecommunication and Information Technology, Vol:3, pp 30-36, ISSN-1509-4553, (2002)
2. Schuster A., Adamson K., Bell D.A. Problem-solving in a Self-similar World and Adaptive Fuzzy Sets. Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, pp. 193-196, Honolulu, Hawaii, USA, (1999)
3. Schuster, A., Adamson, K., Bell, D.A. An Application of a Genetic Algorithm for Rule Weight Optimisation in a Fuzzy Expert System. Proceedings of IASTED International Conference Artificial Intelligence and Soft Computing ASC'99, pp 572-576, Honolulu, USA, (1999)
4. Ross, T.J. Fuzzy Logic with Engineering Applications. McGraw-Hill, New York, London, (1995)
5. Parker, T.S. and Chua, L.O. Chaos: A Tutorial for Engineers. Proceedings of the IEEE, Vol. 75:8, pp. 982-1008, (1987)
6. Lorenz, E.N. Deterministic Non-periodic Flow. J. Atmos. Sc., Vol.20, 130, (1963)
7. Gough, N.A.J. Fractal Analysis of Foetal Heart Rate Variability. Physiol. Meas., 14:309--315, (1993)
8. Grim, P. Self-reference and Chaos in Fuzzy Logic. IEEE Transactions on Fuzzy Systems, Vol.1:4, pp. 237-253, (1993)
9. Elert, G. The chaos hypertextbook. http://hypertextbook.com/chaos, (January 2003).
10. Mandelbrot, B.B. The Fractal Geometry of Nature. Freeman W.H. and Company, New York, (1977)
11. Kaye, B.H. A Random Walk through Fractal Dimensions. Ed. Weinheim, Cambridge, New York, NY, VCH, (1989)
12. Chen, S.J. and Hwang, C.L. Fuzzy Multiple Attribute Decision Making, Methods and Applications. Springer Verlag, Berlin, Heidelberg, (1992)
13. Miller, G.A. The Magic Number Seven, Plus or Minus Two. Psychological Review (63), pp. 81-97, (1965)

A Chromatic Image Understanding System for Lung Cancer Cell Identification Based on Fuzzy Knowledge

Yubin Yang[1], Shifu Chen[2], Hui Lin[1], and Yukun Ye[3]

[1] Joint Laboratory for Geoinformation Science, The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong
{yangyubin, huilin}@cuhk.edu.hk
[2] State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing 210093, P. R.China
chensf@nju.edu.cn
[3] Nanjing Bayi Hospital, Nanjing 210002, P. R. China

Abstract. This paper presents an intelligent medical chromatic image understanding system for lung cancer cell identification based on fuzzy knowledge representation and reasoning. Following image analysis and a low-level feature extraction process, a two-layer rule-based fuzzy knowledge model is proposed to represent the domain knowledge needed for image understanding task. Experimental results show that the system achieves not only a high rate of overall correct identification, but also a low rate of false negative identification, that is, a low rate of identifying cancer cases to be normal ones, which is important in reducing false diagnosis cases.

1 Introduction

Lung cancer is one of the most common and deadly diseases in the world. Detection of lung cancer in early stage is crucial for its cure. In general, measures for early stage lung cancer diagnosis mainly includes those utilizing X-ray chest films, CT, MRI, isotope, bronchoscopy, etc., in which a very important measure is the so-called pathological diagnosis which analyzes the needle biopsy specimens obtained from the bodies of the subjects to be diagnosed [1]. Presently, the needle biopsy specimens are usually analyzed by experienced pathologists. Since senior pathologists are rare, reliable pathological diagnosis is not always available.

In past decades, with the rapid development of image processing and pattern recognition techniques, lots of image understanding systems involving a wide range of domains had been built, and many research achievements had been attained as well. Particularly, medical image understanding for computer-aided diagnosis hace been always attracting more and more attentions [2]. Examples are as follows. Kummert et al. [3] implemented a medical image understanding system named ERNEST, which can understand the Scintigraphy and MRI images. Tombropoulos et al. [4] presented their research on focus diagnosis of liver's MRI images based on belief network. Alireza Osareh et al. [5] developed a system to understand retina

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 392-401, 2004.

images using FCM algorithm and neural network. Related research work has also indicated that it is insufficient only using image processing algorithms for medical image understanding purposes, without related domain knowledge introduced to guide the process [6].

In this paper, we propose a novel image understanding method to identify lung cancer cells in the chromatic images of microscope slices of needle biopsy specimens, which performs intelligent understanding of lung cancer images by integrating image processing, image analysis techniques and a two-layer rule-based fuzzy knowledge model. The knowledge model is built and represented as domain-specific fuzzy rule sets, by which our method achieves not only a high rate of overall correct identification, but also a low rate of false negative identification, that is, a low rate of identifying cancer cases to be normal ones, which is important in reducing false diagnosis cases. The knowledge model has now been successfully implemented in a lung cancer image understanding application named Lung Cancer Diagnosis System (LCDS).

The rest of this paper is organized as follows. Firstly, the framework of LCDS is shortly described in Section 2. Next, Section 3 presents the process of low-level image feature extraction. The rule-based fuzzy knowledge model is proposed in Section 4 and experimental results are presented in Section 5. Finally, conclusion remarks and future work issues are given in Section 6.

2 Framework

The framework of the image understanding system LCDS is depicted in Fig. 1. As illustrated in the left part of Fig. 1, the hardware configuration of LCDS mainly includes a medical electron microscope whose amplification capacity is 400, a digital video camera, an image capturer, and the output devices including printer and video display. The digital video camera is mounted on the electron microscope to take the video signals of the haematoxylin-eosine (HE) stained specimens of needle biopsies. Those video frames are captured by the image capturer and then saved as 24 bit RGB color images, on the basis of which the computer-aided lung cancer image understanding software will identify lung cancer cells automatically according to its diagnosing flow shown in the right part of Fig. 1.

The complete lung cancer cell identification flow of our computer-aided image understanding software is described as follows: firstly, after a RGB image is captured, it is projected from three-dimensional RGB color space into a one dimensional gray level space in 256 scales using a customized algorithm for lung cancer images [7]. Then, image processing techniques including smoothing, contrast enhancement and color enhancement are utilized to improve the image's quality. After that, the image is thresholded and we use the chain code representation [8] to mark all the possible cells in the image. At the same time, RGB information of all the possible cells and the whole image is simultaneously kept for later use. Then, shape and chromatic features of all the possible cells are extracted to feed in a two-layer fuzzy knowledge model, which is established according to diagnosing experience of medical experts, in order to identify whether lung cancer cells exist in the specimen or not.

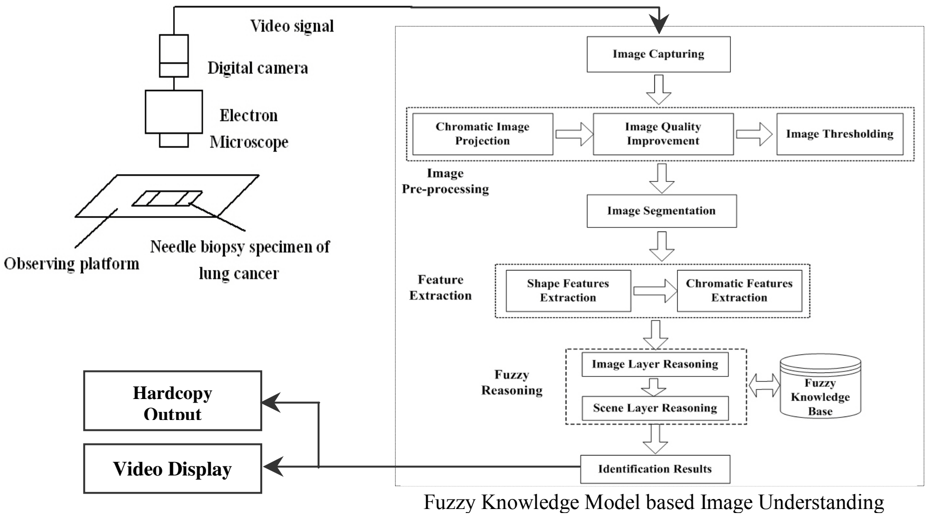


Fig. 1. The Framework of Lung Cancer Diagnosis System (LCDS)

3 Low-Level Feature Extraction

Generally, experienced pathologists identify lung cancer cell according to a number of representative characteristics of its appearance, such as "big nucleolus", "irregular shape", "darker color", and "rough and non-uniform chromatin distribution", etc. [9] Based on those kinds of practical medical experience, we can assume that a lung cancer cell consists of two types of feature: shape and chromatic features. We then combine both shape and chromatic features of each marked cell as a feature vector. Thus, an image can be treated as a collection of cells, each of which is represented by a feature vector, capturing the image's content in shape and chromatic aspects. Those feature vectors are then viewed as a start-up of fuzzy knowledge representation.

3.1 Shape Feature Extraction

The chain code method is a popular and efficient way for contour coding, and it has given raise to a variety of research results and applications on shape description [10]. In LCDS, the shape features are computed based on 8-connect chain code representation [8], which are described as follows:

(1) Cell Circumference

The cell circumference is the length of chain code, which is computed as follows:

$$L = n_e + \sqrt{2} * n_o \quad (1)$$

where n_e is the number of even values (e=0,2,4,6) in the cell's chain code sequence, n_o is the number of odd values (o=1,3,5,7) in the cell's chain code sequence.

(2) Cell Area

For each direction i (i=0,1,…,7) in 8-connect chain code, we define $x(i)$ as its component along X axis and $y(i)$ as its component along Y axis. So, the values of $x(i)$ and $y(i)$ for each direction i are shown in Table 1.

Table 1. Values of $x(i)$, $y(i)$

i	$x(i)$	$y(i)$
0	1	0
1	1	1
2	0	1
3	–1	1
4	–1	0
5	–1	–1
6	0	–1
7	1	–1

The cell area, i.e., the integral of its chain code along X axis, then can be computed as follows:

$$S = \sum_{j=1}^{n} x(a_j) \cdot (y_{j-1} + \frac{1}{2} y(a_j)) \tag{2}$$

where n is the number of values in the cell's chain code sequence, a_j(1 j n) is the chain code value of the cell's edge, y_0 is the Y-coordinate value of the start pixel of chain code sequence, and we have $y_j = y_{j-1} + y(a_j)$.

(3) Cell Width

As a measurement for shape abnormality, the cell width is determined by the following equation:

$$W = \max_{n} (\sum_{j=1}^{n} x(a_j) + x_0) - \min_{n} (\sum_{j=1}^{n} x(a_j) + x_0) \tag{3}$$

Here x_0 is the X-coordinate value of the start pixel of chain code sequence.

(4) Cell Height

The cell height is computed by the following equation:

$$H = \max_{n} (\sum_{j=1}^{n} y(a_j) + y_0) - \min_{n} (\sum_{j=1}^{n} y(a_j) + y_0) \tag{4}$$

Here y_0 is the Y-coordinate value of the start pixel of chain code sequence.

(5) Roundness Degree

The roundness degree is introduced to represent the cell's deviation from roundness, which is given as:

$$C = \frac{4\pi \bullet (Area)}{(Circumfere\ nce)^2} = \frac{4\pi \bullet S}{L^2} \tag{5}$$

Given the same size, a round and boundary-smooth cell should have the shortest circumference, whose roundness degree equals 1. The more the cell deviates from roundness, the less roundness degree value it has.

(6) Protraction Degree

The narrower the cell is, the less protraction degree value it has. It is defined as:

$$E = \frac{\min(Width, Height)}{\max(Width, Height)} = \frac{\min(W, H)}{\max(W, H)} \tag{6}$$

When a cell is totally round, it has the max protraction degree value 1.

3.2 Chromatic Feature Extraction

Since there exist multiple color space representations [11], each of which describes different aspects of colors, we need choose those color components which can best describe color characteristics of lung cancer cells. The original images are RGB color image, so we can get the RGB component in a nature way, through which other chromatic features can also be computed.

We have utilized the following color components as features used for lung cancer cell identification.

(1) R, G, B components of cell

Firstly, RGB color space is selected for chromatic feature extraction. Here we compute the mean and variance of Red, Green and Blue component of all the pixels in a marked cell as its representative features.

(2) H, I, S components of cell

We also choose HIS color space for chromatic feature extraction, because it reflects lung cancer cell's perceptive characteristics better. The conversion from RGB to HLS is defined as follows [12]:

$$H = \cos^{-1}\left\{\frac{(2R - G - B)/2}{\left[(R - G)^2 + (R - B)(G - B)\right]^{1/2}}\right\} \tag{7}$$

$$I = \frac{R + G + B}{3} \tag{8}$$

$$S = 1 - \frac{3\min(R, G, B)}{R + G + B} \tag{9}$$

where *H* is Hue, *I* is Intensity and *S* is Saturation.

The mean and variance of Hue, Intensity and Saturation component of all the pixels in a marked cell are then computed as its representative features.

(3) Self-defined C_{ratio} component of cell

The computation is defined as Eq. (10):

$$C_{ratio} = B/(R+G+B) \tag{10}$$

The introduced C_{ratio} component is an important characteristic for describing the lung cancer cells. It is derived from the medical fact that lung cancer cells are always blue and purple to some extent, which is proved to affect the identification result a lot.

(4) R, G, B components of image

The identification task based on image content should be adaptive to slight variation of image colors, so we also adopt the mean value of RGB components of the whole slice image as identification features, which enhances the robustness and flexibility of identification task well.

4 Rule-Based Fuzzy Knowledge Model

The task of image understanding is to identify objects in an image using the relevant knowledge. So, it is necessary to set up a model for representing domain knowledge representation [6]. To lung cancer image understanding, it is especially the most crucial and challenging issue how to represent the domain knowledge in a form that the computer can deal with. To solve the problem, a hierarchical rule-based fuzzy knowledge model is designed for LCDS, by which the mapping relation from the low-level image features to the scene descriptor of lung cancer images is implemented.

4.1 Mathematical Model for Fuzzy Knowledge

In knowledge-based image understanding system, different types of knowledge are required for different processing steps. We design an intermediate layer, i.e., scene feature layer, as a bridge connecting the raw image data and the lung cancer image descriptors. The whole description set of lung cancer image is then composed of three layers, which are respectively defined as follows:

Def. 1 Low-level feature set: $\boldsymbol{\Omega}= \{\boldsymbol{\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(m)}}\}$, which denotes the extracted low-level image features, including both shape features and chromatic features.

Def. 2 Scene feature set: $\boldsymbol{\Psi}= \{\boldsymbol{\psi^{(1)}, \psi^{(2)}, \ldots, \psi^{(n)}}\}$, which denotes visually perceptive description provided by domain experts, such as $\psi^{(i)}$={irregular shape}, $\psi^{(j)}$={non-uniform chromatin distribution}.

Def. 3 Scene descriptor set: $\boldsymbol{£}= \{\boldsymbol{\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(k)}}\}$, which denotes all the possible identification results, such as $\boldsymbol{\tau}^{(i)}$={cancer cell},$\boldsymbol{\tau}^{(j)}$={abnormal cell}.

Thus, by introducing scene feature set $\boldsymbol{\Psi}$, we can establish two mapping functions as follows:

(1) $f(\omega^{(i)}, \psi^{(j)})$: $\quad \boldsymbol{\Omega} \text{ x } \boldsymbol{\Psi} \rightarrow \mathbf{R_1}$

(2) $f'(\psi^{(l)}, \tau^{(k)})$: $\boldsymbol{\Psi} \text{ x } \boldsymbol{£} \rightarrow \mathbf{R_2}$

In LCDS, the above two mapping functions defines two fuzzy relations, that is, fuzzy relation R_1 from low-level image feature set $\boldsymbol{\Omega}$ into scene feature set $\boldsymbol{\Psi}$, and relation R_2 from scene feature set $\boldsymbol{\Psi}$ into scene descriptor set $\boldsymbol{£}$. Note that R_1 is a fuzzy subset of the Cartesian product $\boldsymbol{\Omega}$ x $\boldsymbol{\Psi}$ and R_2 is a fuzzy subset of the Cartesian product $\boldsymbol{\Psi}$ x $\boldsymbol{£}$. Both R_1 and R_2 can be represented by a matrix with coefficients in the interval [0,1], where the general entry $R_1(x_1,y_1)$ is defined by the membership of (x_1,y_1) in R_1, for all $x_1 \in \boldsymbol{\Omega}$ and $y_1 \in \boldsymbol{\Psi}$, and the general entry $R_2(x_2,y_2)$ is defined by the membership of (x_2,y_2) in R_2, for all $x_2 \in \boldsymbol{\Psi}$ and $y_2 \in \boldsymbol{£}$. All these membership values are summarized from the diagnosis experience given by medical experts, which quantify the association degrees of different features in lung cancer diagnosis process.

4.2 Fuzzy Knowledge Representation

In LCDS, the two fuzzy relations in the knowledge model, i.e. R_1 and R_2, are both implemented using fuzzy rules and uncertainty reasoning based on Fuzzy Theory. Fuzzy Theory originates from the mathematic foundation of Zadeh's Fuzzy Set Theory, firstly proposed by Zadeh in 1965, including Fuzzy Set Theory, Fuzzy Logic, Fuzzy Reasoning and Fuzzy Control Theory etc. It is fully developed and widely applied after Zadeh's report on Fuzzy Reasoning [13]. Fuzzy Logic and Fuzzy Reasoning have already had many successful applications on image processing research, especially for the complicated images that can be hardly defined using mathematic models. Medical image understanding problem is a good example [14], [15]. Medical Experts always describe their diagnosis experience in the form of natural language, which is an uncertain process but have to be implemented using non-linear algorithms. Fuzzy theory is a suitable way to represent and explain those fuzzy and uncertain medical concepts.

In LCDS, the medical expert's control strategies on lung cancer diagnosis are settleed and represented as two kinds of knowledge, that is, image layer knowledge and scene layer knowledge, implemented using production rules in the form of "IF (Antecedent) THEN (Consequent)". Considering the uncertainty processing, we define the fuzzy rule in the following form:

IF *(X is A_j)* AND *(Y is B_j)* THEN (*Z is C_j*) M_LEVEL: (l_{Aj}, l_{Bj})
WITH PARAMS (ω_{Aj}, ω_{Bj}, ω_{Cj})

X and *Y* denote the input variables. For fuzzy relation R_1, they are low-level image features, while for fuzzy relation R_2 they are scene features. A_j is the linguistic or fuzzy value assigned over *X* while B_j is that assigned over *Y*. *Z* is the output variable, which is scene feature for R_1 and scene descriptor for R_2. C_j is the fuzzy set assigned over *Z*, l_{Aj} and l_{Bj} are the lowest membership thresholds of A_j and B_j to fire the rule. ω_{Aj}, ω_{Bj} are the corresponding elements in the membership matrix of fuzzy relation R_1 or R_2, and ω_{Cj} is pre-defined normalized "rule membership" to describe the importance of each rule.

The inference operator is then presented as:

$$C_j(Z) = \omega_{C_j} \bullet (\omega_{A_j} \bullet A_j(x_0) + \omega_{B_j} \bullet B_j(y_0)) \tag{11}$$

where $C_j(Z)$ is the output fuzzy value of the above rule for each crisp input pair (x_0,y_0).

With the above fuzzy inference operation in our rule base, we can get a fuzzy subset of scene descriptor set ***£***. The final lung cancer image understanding results can be made according to an *argmax* operator applied on this subset:

$$C = \arg\max_{C_j \in £} (\sum C_j(Z)) \tag{12}$$

where Z denotes the final output variables, namely, the lung cancer cell identification results.

4.2.1 Image Layer Knowledge

The image layer knowledge corresponds to the fuzzy relation R_1. This layer aims at reasoning scene features at intermediate level from the extracted low-level image features. The antecedent part of image layer knowledge is a logic combination of low-level image features, while its consequent part is a scene feature. In fact, image layer knowledge provides the visual properties of cells in an image.

In image layer, a "bottom-up" reasoning strategy is adopted to transform the representation of image from pixel level into an intermediate level, which will then fire the reasoning process in scene layer.

The following is an instance of image layer knowledge in LCDS:

IF (The cell is LARGE) and (The cell is NOT QUITE ROUND) M_LEVEL: (0.3,0.7)
THEN (The shape abnormity of cell is MEDIUM)
WITH PARAMS (0.3,0.55,0.4)

4.2.2 Scene Layer Knowledge

The scene layer knowledge is used for high-level reasoning in image understanding task, i.e., the final identification of lung cancer cells, corresponding to the fuzzy relation R_2. The antecedent part of the scene layer knowledge is a logic combination of scene features, while its consequent part is a scene descriptor, which is the final lung cancer cell identification result.

In scene layer, the reasoning strategy is switched to a "Top-down" way, validating or rejecting the hypothesis on scene descriptors according to the output fuzzy values deduced from the facts on scene features.

The following is an instance of scene layer knowledge in LCDS:

IF (The cell is in abnormal shape) M_LEVEL: 0.6
THEN (The cell is identified as lung cancer cell possibly)
WITH PARAMS (0.42,0.45)

5 Experimental Results

We have used 255 chromatic images from 119 pieces of microscope slice, 2~3 images for each, from year 1996 to year 1999 to evaluate the image understanding performance of our method. All the samples are well-archived real cases ever diagnosed by medical experts of Nanjing Bayi Hospital, whose diagnosis results are stable and undisputable, which makes our experimental results comparable and significative. In Table 2, the experimental results are listed. There are three error measures. *Error Rate* measures the rate of overall false identification computed by dividing the number of false identified images by the number of test images. *False Negative* measures the rate of false negative identification computed by dividing the number of images, which are cancer cases but erroneously identified as normal cases, by the number of test images. *False Positive* measures the rate of false positive identification computed through dividing the number of images, which are normal cases but erroneously identified as cancer cases, by the number of test images.

The above results approve the validity of the two-layer rule-based fuzzy knowledge model proposed by this paper from several aspects. Firstly, in real cases, even for experienced medical experts there exists unavoidable possibility to conduct false diagnosis. The *Error Rate* of LCDS is quite acceptable in practice as a fast, laborsaving and preliminary diagnosing means. Especially for 1999, in which there exists less depigmentation of slice than other three years, the *Error Rate* of LCDS is very close to that of medical experts. Secondly, for lung cancer diagnosis, *False Negative* should be strictly controlled even at the cost of increasing *False Positive*, because diagnosing a lung cancer patient to be healthy is a very serious mistake that will possibly result in the loss of life. As shown in Table 2, the most crucial improvement of our method is that the average *False Negative* of LCDS is well controlled at an approving level. Meanwhile, the *False Positive* does not increase too much, keeping at a tolerable level. This performance to large extent guarantees its practicability in real diagnosis. Accordingly, LCDS is very effective and efficient for lung cancer image understanding task, which aids medical experts to identify lung cancer simply and fast.

Table 2. Experimental results of lung cancer image understanding in LCDS

Year of Slices	Number of Slices	Num of Images	False Negative	False Positive	Error Rate
1996	21	39	2.6%	7.7%	10.3%
1997	26	65	3.1%	12.3%	15.4%
1998	33	51	7.8%	3.9%	11.7%
1999	39	100	1.0%	5.0%	6.0%

However, it can also be clearly learned that there still has potential to further improve the identification results. The knowledge model should be adjusted more specifically and adaptively according to the question field. Moreover, how to improve the robustness against slice contamination is also another issue requiring more considerations.

6 Conclusions

This paper presents an intelligent medical chromatic image understanding system for lung cancer cell identification based on a two-layer rule-based fuzzy knowledge model. Experimental results show that the system achieves good image understanding performance by identifying lung cancer cells in the images efficiently and effectively. We believe that improving the knowledge model more specifically in the domain, the performance of the identification process will be much more advanced.

We also believe that improving the knowledge representation technique is just one aspect of LCDS, the performance of the identification process will be much more enhanceed if we can make a betterment on image processing techniques such as chromatic image segmentation. Still, we hope to elaborate the fuzzy knowledge in more details to make it capable of classifying different types of lung cancer as well.

Acknowledgements. This research has been funded in part by the National Natural Science Foundation of P. R. China under grant No.39670714 and the National 9th Five-year Planning of P. R. China under grant No.96-906-01-18. We thank the collaborators in Nanjing Bayi hospital for their suggestive discussion on the domain-specific knowledge.

References

1. Ye Y.K., Shao C., Ge X.Z. et al.: Design and Clinical research of a Novel Instrument for Lung Cancer Early-stage Diagnosis. Chinese Journal of Surgery, 30 (5) (1992) 303-305
2. Bezdek J.C., Hall L.O., Clarke L.P.: Review of MR image segmentation techniques using pattern recognition. Medical Physics 20 (4) (1993) 1033-1048
3. Kummert F. et al.: Control and explanation in a signal understanding environment. Signal Processing 32 (1993) 111-145
4. Tombropoulos R. et al.: A Decision Aid for Diagnosis of Liver Lesions on MRI. Section on Medical Informatics, Stanford University School of Medicine, AMIA (1994)
5. Osareh A., Mirmehdi M., Thomas B., Markham R.: Automatic recognition of exudative maculopathy using fuzzy c-means clustering and neural networks. In: Medical Image Underestanding and Analysis Conference. BMVA Press (2001) 49-52
6. Grimson W.: Medical Applications of Image Understanding. IEEE Expert 10(1995) 18-28
7. Yang Y.B., Li N., Chen S.F., Chen Z.Q.: Lung Cancer Identification Based on Image Content. In: 6th International Conference for Young Computer Scientists. 1 (2001) 237-240
8. Freeman H.: Computer Processing of Line-drawing Image. Computing Surveys 6 (1) (1974) 57-97
9. Section of Pathology, Shanghai Tumor Hospital: Applied Tumor Cytology. Shanghai, Shanghai People's Publishing House (1975)
10. Baruch O., Loew M. H.: Segmentation of Two-dimensional Boundaries Using the Chain Code. Pattern Recognition 21 (6) (1988) 581-589
11. Gaurav S.: Digital Color Imaging. IEEE Transactions On Image Processing 6 (7) (1997) 901-932
12. Mehtre B.M., Kankanhalli M.S., et al.: Color Matching for Image Retrieval. Pattern Recognition Letters 16 (3) (1995) 325-331
13. Zadeh L.A.: Fuzzy Logic and its Application to Approximate Reasoning. Information Processing 74 (1974) 591-594
14. Terano T., Asai K., Sugeno M.: Fuzzy System Theory and its Applications. Boston, Academic Press (1992)
15. Kuncheva L.I.: How Good Are fuzzy If-then Classiers?. IEEE Transactions on Systems, Man, and Cybernetics 30 (4) (2000) 501-509

Reading Human Relationships from Their Interaction with an Interactive Humanoid Robot

Takayuki Kanda[1] and Hiroshi Ishiguro[1,2]

[1] ATR, Intelligent Robotics and Communication Laboratories
2-2-2 Hikaridai Sorakugun Seikacho, 6190288 Kyoto, Japan
kanda@atr.co.jp
http://www.irc.atr.co.jp/

[2] Osaka University, Graduate School of Engineering, Dept. Adaptive Machine Systems
Osaka, Japan
ishiguro@ams.eng.osaka-u.ac.jp

Abstract. This paper reports our novel approach to developing a social robot. Such a robot reads human relationships from their physical behavior. We have developed an interactive humanoid robot that attracts humans to interact with it and, as a result, induces their group behaviors in front of it. In our approach, the robot recognizes friendly relationships among humans by simultaneously identifying each person in the interacting group. We conducted a two-week experiment in an elementary school, in which Robovie demonstrated proven reasonable performance in identifying friendships among the children. We believe this ability to read human relationships is essential to behaving socially.

1 Introduction

Recent progress in robotics has brought with it a new research direction known as “interaction-oriented robots.” These robots are different to traditional task-oriented robots, such as industrial robots, which perform certain tasks in limited applications. Interaction-oriented robots are designed to communicate with humans and to be able to participate in human society. We are trying to develop such an interaction-oriented robot that will exist as a partner in people’s daily lives. As well as providing physical support, these robots will supply communication support such as route-guidance.

Several researchers are endeavoring to realize such interaction-oriented robots. Aibo has become the first interactive robot to prove successful on the commercial market [1], since it behaves as if it were a real animal pet. Breazeal et al. developed the face robot Kismet, and they are exploring sociable aspects of robots produced through its learning ability [2]. Okuno et al. developed a humanoid head that tracks a speaking person with visual and auditory data. In addition, they controlled the personality of the robot by changing the tracking parameter [3]. Burgard et al. developed a museum tour guide robot [4] that was equipped with robust navigational skills and behaved as a museum orientation tool. These research efforts also seem to be devoted to social robots that are embedded in human society.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 402-412, 2004.

Humans have the natural ability to read others' intentions, which is widely known as the joint-attention mechanism in developmental psychology [5], and we believe that this is an essential function for both humans and robots to be social. Scassellati developed a robot with a joint-attention mechanism that follows others' gazes in order to share attention [6]. Kozima et al. also developed a robot with a joint-attention mechanism [7]. In other words, these robots read humans' intentions from their behaviors. Furthermore, a robot system can estimate humans' subjective evaluation of it by observing the humans' body movements [8]. However, these research works mainly focused on the social behaviors among two or three people. Little robotics research work has been attempted to handle social behavior within greater human society. To enable a robot to be social, we believe it is important for the robot to read relationships among humans.

In sociology, sociometric (a matrix that represents relationships) and socio-gram (a direct graph that illustrates the sociometric) methods have been used to represent the relationships that occur among humans. A sociometric test is a subjective test that retrieves the relationships, and lets a human directly answer the name of others whom he/she likes and dislikes. It has been widely used to determine the relationships in a classroom or a company; however, it has recently become difficult to apply (in particular, asking the name of disliked persons), since it might promote negative relationships.

In the field of computer science, several research works have analyzed human relationships. Eveland et al. analyzed online communication on a CSCW (computer-supported collaborative work) system [9]. They plotted each user's data on a socio-gram according to the amount of online communication among them. Nomura and her colleagues developed a Web-analyzing system to retrieve humans' online relationships from hyperlinks in their web pages [10]. Watts and Strogatz conducted a computer simulation to find a simple model for global human society and proposed that small-world networks represent large-scale human relationships [11].

In this paper, we report our approach to reading human relationships with an interactive robot, an ability that is probably essential for interactive robots to be social. We have developed an interactive humanoid robot, named Robovie, that autonomously interacts with humans. As a result, the robot attracts humans to interact with it and induces humans' group behaviors in front of it. In our approach, the robot recognizes friendly relationships among humans by simultaneously identifying each person in the interacting group. We conducted a two-week experiment in an elementary school, in which Robovie demonstrated reasonable performance in identifying friendships among the children.

2 Robovie: An Interactive Humanoid Robot

2.1 Hardware

Figure 1 shows the humanoid robot "Robovie" [12]. The robot is capable of human-like expression and recognizes individuals by using various actuators and sensors. Its body possesses highly articulated arms, eyes, and a head, which were designed to produce sufficient gestures to communicate effectively with humans. The sensory equipment includes auditory, tactile, ultrasonic, and vision sensors, which allow the robot to behave autonomously and to interact with humans. All processing and control systems, such as the computer and motor control hardware, are located inside the robot's body.

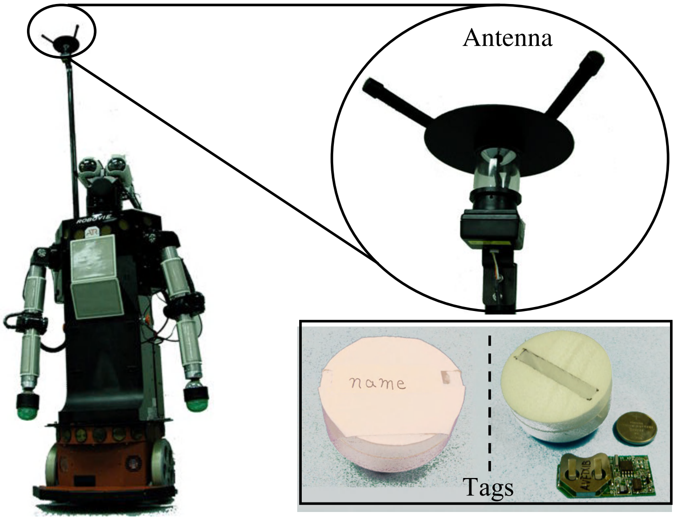


Fig. 1. Robovie (left) and Wireless tag. *Robovie is an interactive humanoid robot that autonomously speaks, makes gestures, and moves around. With its antenna and tags, it is able to identify individuals.*

2.2 Person Identification with Wireless ID Tags

To identify individuals, we used a wireless tag system capable of multi-person identification by partner robots (Detailed specification and system configuration is described in [13]). Recent RFID (radio frequency identification) technologies have enabled us to use contact-less identification cards in practical situations. In this study, children were given easy-to-wear nameplates (5 cm in diameter) in which a wireless tag was embedded. A tag (Fig. 1, lower-right) periodically transmitted its ID to the reader installed on the robot. In turn, the reader relayed received IDs to the robot's software system. It was possible to adjust the reception range of the receiver's tag in real-time by software. The wireless tag system provided the robots with a robust means of identifying many children simultaneously. Consequently, the robots could show some human-like adaptation by recalling the interaction history of a given person.

2.3 Interactive Behaviors

"Robovie" features a software mechanism for performing consistent interactive behaviors (detailed mechanism is described in [14]). The objective behind the design of Robovie is that it should communicate at a young child's level. One hundred interactive behaviors have been developed. Seventy of them are interactive behaviors such as shaking hands, hugging, playing paper-scissors-rock, exercising, greeting, kissing, singing, briefly conversing, and pointing to an object in the surroundings. Twenty are idle behaviors such as scratching the head or folding the arms, and the remaining 10 are moving-around behaviors. In total, the robot could utter more than 300 sentences and recognize about 50 words.

Several interactive behaviors depended on the person identification function. For example, there was an interactive behavior in which the robot called a child's name if that child was at a certain distance. This behavior was useful for encouraging the child to come and interact with the robot. Another interactive behavior was a body-part game, where the robot asked a child to touch a body part by saying the part's name.

The interactive behaviors appeared in the following manner based on some simple rules. The robot sometimes triggered the interaction with a child by saying "Let's play, touch me," and it exhibited idling or moving-around behaviors until the child responded; once the child reacted, it continued performing friendly behaviors for as long as the child responded. When the child stopped reacting, the robot stopped the friendly behaviors, said "good bye," and re-started its idling or moving-around behaviors.

3 Reading Humans' Friendly Relationships

3.1 Basic Ideas

Our approach consists of the two functions described below (Figure 2). Since humans have friendly relationships, they behave in a group. Meanwhile, a robot induces spontaneous group behavior with its interactive behaviors.

3.1.1 Group Behavior and Friendship

Like and *dislike* are two of the essential relationships among humans. Humans change their opinions based on *like* and *dislike* relationships, which is well-known as Heider's balance theory [15]. For example, if a person's friend has an opposing opinion, the person would change his/her opinion to be agreeable with the friend's opinion. Humans establish friendship based on their mutual "like" relationships of each other. In developmental psychology, Ladd et al. found that even children form their own group and behave with the group based on their friendship [16]. In other words, if we observe such a group's behavior, we can estimate the friendships among the members.

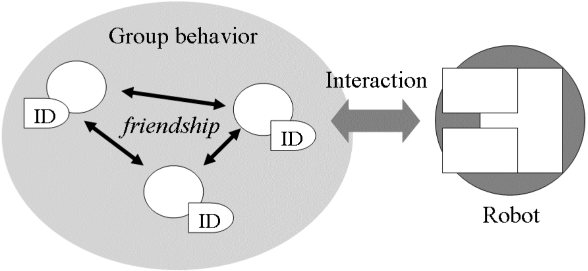


Fig. 2. Mechanism of reading humans' friendly relationships. *Robot identifies multiple people in front of it simultaneously; as a result, it recognizes friendship among them, because the robot's interactive behaviors cause the group behavior.*

3.1.2 Use of Interactive Robot to Cause Spontaneous Group Behavior

Our interactive humanoid robot Robovie autonomously interacts with humans. By executing interactive behaviors, the robot attracts humans to interact with it; on the other hand, humans often behave in a group, so the robot induces human group behaviors in front of it. As a result, the robot can recognize friendly relationships among humans by simultaneously identifying each person in the interacting group.

We might read such friendly relationships by simply observing humans' group behavior in their daily life. However, humans sometimes behave as a group because it is necessary or required. For example, the activity "humans collaborate to carry a heavy box" does not always indicate friendly relationships among them. Thus, we believe that it is better to read human relationships by observing spontaneous group behavior such as interaction with the robot. We believe that in the future robots will carry out various communication tasks in our daily lives such as foreign language education [17], and humans will freely interact with robots even in these applications.

3.2 Algorithm

From a sensor (in this case, wireless ID tags and receiver), the robot constantly obtains the IDs (identifiers) of individuals who are in front of it. The robot continuously accumulates its interacting time with person A (T_A) and the time that person A and B simultaneously interact with it (T_{AB}, which is equivalent to T_{BA}). We define the estimated friendship from person A to B ($Friend(A \to B)$) as

$$Friend(A \to B) = if\,(T_{AB} / T_A > T_{TH}), \tag{1}$$

$$T_A = \Sigma\; if\,(observe(A) \text{ and } (S_t < S_{TH})\,) \cdot \Delta t, \tag{2}$$

$$T_{AB} = \Sigma\; if\,(observe(A) \text{ and } observe(B) \text{ and } (S_t < S_{TH})\,) \cdot \Delta t\,, \tag{3}$$

where *observe*(A) becomes true only when the robot observes the ID of person A, *if*() becomes 1 when the logical equation inside the bracket is true (otherwise 0), and T_{TH} is a threshold of simultaneous interaction time. We also prepared a threshold S_{TH}, and the robot only accumulates T_A and T_{AB} so that the number of persons simultaneously interacting at time t (S_t) is less than S_{TH} (Eqs. 2 and 3). In our trial, we set Δt to one second.

4 Experiment

We conducted a field experiment in an elementary school for two weeks with the developed interactive humanoid robot, which was originally designed to promote children's English learning. As we reported in [17], the robots had a positive affect on the children. In this paper, we use the interaction data during that trial as a test-set of our approach to reading friendship from the children's interaction.

4.1 Method

We performed an experiment at an elementary school in Japan for two weeks. Subjects were sixth-grade students from three different classes, totaling 109 students (11-12 years old, 53 male and 56 female). There were nine school days included in those two weeks. Two identical robots were placed in a corridor that connects the three classrooms (Figure 3). Children could freely interact with both robots during recesses (in total, about an hour per day), and each child had a nameplate with an embedded wireless tag so that each robot could identify the child during interaction.

We administered a questionnaire that asked the children to write down the names of their friends. This obtained friendship information was collected for comparison with the friendship relationships estimated by our proposed method.

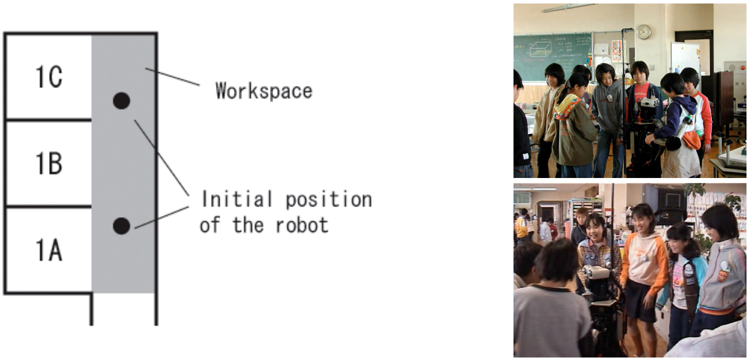


Fig. 3. Environment of the elementary school where we installed the robot. *Left figure is a map of the environment, and right photos are scenes of the experiment.*

Table 1: Estimation results with various parameters

coverage reliability		*T_{TH} (simultaneously interacting time)*					
		0.3	0.2	0.1	0.05	0.01	0.001
S_{TH} (num. of simultane-ously Interacting children)	2	0.01	0.02	0.03	0.04	0.04	0.04
		1.00	0.93	0.79	0.59	0.54	0.54
	5	0.00	0.02	0.06	0.11	0.18	0.18
		1.00	1.00	0.74	0.47	0.29	0.28
	10	0.00	0.00	0.04	0.13	0.29	0.31
		-	1.00	0.74	0.46	0.23	0.20

('-' indicates that no relationships were estimated, so reliability was not calculated)

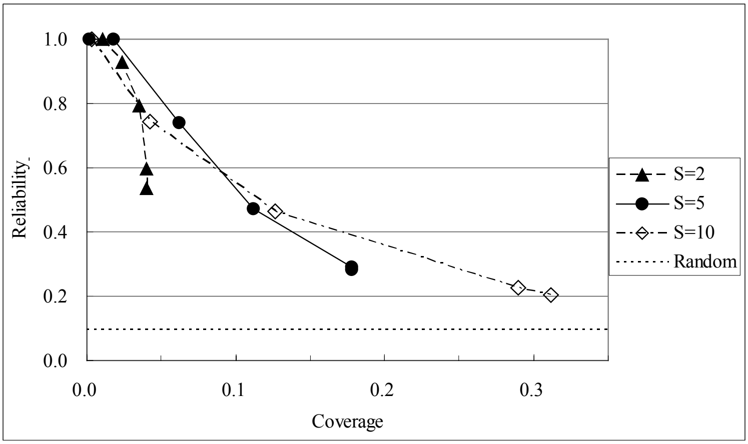


Fig. 4. Illustrated estimation results with various parameters. *(Each line corresponds with the S_{TH} (2, 5, and 10). Each point of these lines corresponds with a certain T_{TH} in Table 1.)*

4.2 Results

4.2.1 Evaluation of Estimated Relationships with Reliability and Coverage

Since the number of friendships among children was fairly small, we focused on the appropriateness (coverage and reliability) of the estimated relationships. This is similar to the evaluation of an information retrieval technique such as a Web search. Questionnaire responses indicated 1,092 friendships among a total of 11,772 relationships; thus, if we suppose that the classifier always classifies a relationship as a non-friendship, it would obtain 90.7% correct answers, which means the evaluation is completely useless. Thus, we evaluate our estimation of friendship based on reliability and coverage, which are defined as follows.

Reliability = number of correct friendships in estimated friendships / number of estimated friendships

Coverage = number of correct friendships in estimated friendship / number of friendships from the questionnaire

Table 2. Gender effect (at S=5)

reliability	T_{TH}					
	0.3	0.2	0.1	0.05	0.01	0.001
Male-Male	-	1.00	0.81	0.60	0.45	0.44
Female-Female	1.00	1.00	0.80	0.61	0.43	0.41
Male-Female	-	-	0.20	0.03	0.01	0.01

('-' indicates that no relationships were estimated, so reliability was not calculated)

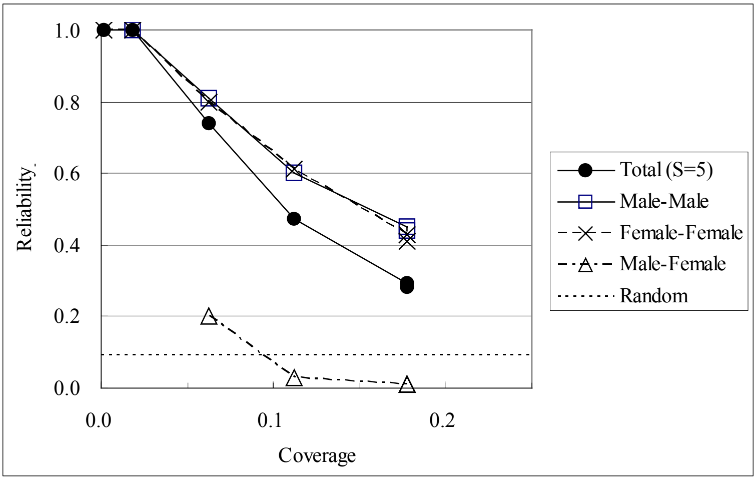


Fig. 5. Illustration of gender effects at S_{TH} =5. *(Each point of these lines corresponds with a certain T_{TH} in Table 2. The plotted coverage of all groups is based on the coverage of "total" group, which is equal to "S_{TH} =5" in Table 1.)*

Table 1 and Fig. 4 indicate the results of estimation with various parameters (S_{TH} and T_{TH}). In Fig. 4, *random* represents the reliability of random estimation where we assume that all relationships are friendships (since there are 1,092 correct friendships among 11,772 relationships, the estimation obtains 9.3% reliability with any coverage). In other words, *random* indicates the lower boundary of estimation. Each of the other lines in the figure represents the estimation result with different S_{TH}, which has several points corresponding to different T_{TH}. There is obviously a tradeoff between reliability and coverage, which is controlled by T_{TH} ; S_{TH} has a small effect on the tradeoff, S=5 mostly performs better estimation of the friendship, and S=10 performs better estimation when coverage is more than 0.15. As a result, our method successfully estimated 5% of the friendship relationships with greater than 80% accuracy (at "S=5") and 15% of them with nearly 50% accuracy (at "S=10").

4.2.2 Gender Effects

To verify the appropriateness of the estimation in detail, we analyzed gender effects. We first classified the relationships into three groups: *male-male*, *female-female*, and *male-female* (including both *male to female* and *female to male*). Then we calculated the reliability and coverage for the relationships of the three groups. Table 2 and Fig. 5 show the result at S=5. The *male-male* and *female-female* groups indicate better performance than the *total* case, while the *male-female* group indicated extremely low performance; in fact, the performance of the *male-female* group was lower than *random* performance, because the children rarely reported friendships with the opposite gender. We believe they might hesitate to report friendly relationships with the opposite gender, which would cause the lower performance of our estimation. Meanwhile, this also suggests the positive prospects of our approach because it might be able to estimate friendly relationships that a subjective questionnaire cannot detect.

5 Discussion and Conclusions

We have developed an interactive robot that reads humans' friendly relationships. This was accomplished by the robot inducing friendly group behaviors in front of it and simultaneously identifying multiple people. Experimental results show that our system successfully estimated 5% of the friendly relationships (retrieved by subjective questionnaire) with greater than 80% accuracy, and 15% of them with nearly 50% accuracy. We believe that this is a reasonable performance, although the estimation algorithm is very simple. In addition, the results suggest that the children hesitated to answer whether they had friendly relationships with individuals of the opposite gender, which agrees with our estimation. In other words, in some cases observing the children's interaction probably leads to better acquisition of knowledge about their friendship than a traditional subjective method. We believe that these results demonstrate the promising potential of our approach. This ability of reading human relationships will be essential and useful for social robots. For example, if robots can guess human friendship relationships, they will be able to promote the relationships, join the relationships or reconcile bullying problems.

There was a tradeoff between reliability and coverage of the estimated relationships. The strict threshold for the time when people simultaneously interact resulted in a small amount of estimation with high accuracy, whereas a moderated threshold provided a larger amount of estimation with low accuracy. However, even if we had moderated the threshold, we still could not estimate more than 30% of the friendship relationships. We do not consider this as an upper limitation of estimation. Instead, we believe the amount of data (that is, the interaction between the children and the robot) was not sufficient for all friendships to be observed by the robot. To verify this hypothesis, an important future work is to gather more long-term interaction data as well as to improve the estimation accuracy by finding other effective rules.

Another concern might be the applicability of this approach. In this paper, we applied our interactive robot that behaves like a child to elementary school students. It is not yet clear whether this approach can be extended to general society that includes adults; we do believe, however, that if robots have the ability to keep interacting with adults, they will even be able to estimate adults' friendly relationships, because social behavior during spontaneous interactions seems not to differ between children and adults. For example, they behave in a group based on their friendships. Thus, if we can develop an interactive robot that can interact with adults for a reasonable period of time, such as by giving interesting information and chattering, we can extend our approach to more general society that includes adults.

Acknowledgements. We wish to thank the teachers and students at the elementary school for their agreeable participation and helpful suggestions. We also thank Takayuki Hirano and Daniel Eaton, who helped with this field trial in the elementary school. This research was supported in part by the Telecommunications Advancement Organization of Japan.

References

1. Fujita, M.: AIBO; towards the era of digital creatures, Int. J. of Robotics Research, Vol. 20, No. 10 (2001) 781-794.
2. Breazeal, C. and Scassellati, B.: A context-dependent attention system for a social robot, Proc. Int. Joint Conf. on Artificial Intelligence (1999) 1146-1151.
3. Okuno, H. G., Nakadai, K., and Kitano, H.: Realizing audio-visually triggered ELIZA-like non-verbal behaviors, PRICAI2002, LNAI 2417, Lecture Notes in Artificial Intelligence, Springer-Verlag (2002) 552-562.
4. Burgard, W., Cremers, A. B., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S.: The interactive museum tour-guide robot, Proc. of National Conference on Artificial Intelligence (1998) 11-18.
5. Moore, C. and Dunham, P. J. eds.: Joint Attention: Its Origins and Role in Development, Lawrence Erlbaum Associates (1995).
6. Scassellati, B.: Investigating Models of Social Development Using a Humanoid Robot, Biorobotics, MIT Press (2000).
7. Kozima, H. and Vatikiotis-Bateson, E.: Communicative criteria for processing time/space-varying information, Proc. IEEE Int. Workshop on Robot and Human Communication (2001).
8. Kanda, T., Ishiguro, H., Imai, M., and Ono, T.: Body Movement Analysis of Human-Robot Interaction, International Joint Conference on Artificial Intelligence (2003) 177-182.
9. Eveland, J. D. and Bikson T. K.: Evolving electronic communication networks: an empirical assessment, Proceedings of the 1986 ACM conference on Computer-supported cooperative work (1986) 91-101.
10. Nomura S., Oyama S., Hayamizu T., and Ishida T.: Analysis and Improvement of HITS Algorithm for Detecting Web Communities, The 2002 International Symposium on Applications and the Internet (2002) 132-140.
11. Watts, D. J. and Strogatz, S. H.: Collective dynamics of 'small-world' networks, Nature, Volume 393, Issue 6684 (1998) 440-442.
12. Ishiguro, H., Ono, T., Imai, M., and Kanda, T.: Development of an interactive humanoid robot "Robovie" -An interdisciplinary approach, R. A. Jarvis and A. Zelinsky (eds.), Robotics Research, Springer (2003) 179-191.
13. Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H.: Person Identification and Interaction of Social Robots by Using Wireless Tags, IEEE/RSJ International Conference on Intelligent Robots and Systems (2003).

14. Kanda, T., Ishiguro, H., Imai, M., Ono T., and Mase, K.: A constructive approach for developing interactive humanoid robots, IEEE/RSJ International Conference on Intelligent Robots and Systems (2002) 1265-1270.
15. Heider, F.: The Psychology of interpersonal relations. Wi1ey (1958).
16. Ladd, G. W., Price, J. M., and Hart, C. H.: Preschooler's behavioral orientations and patterns of peer contact: predictive of peer status?, in Asher S. R. and Coie J. D. (eds.) Peer rejection in childhood, Cambridge University Press (1990) 90-115.
17. Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H.: A practical experiment with interactive humanoid robots in a human society, Third IEEE International Conference on Humanoid Robots (2003).

Recognition of Emotional States in Spoken Dialogue with a Robot

Kazunori Komatani, Ryosuke Ito, Tatsuya Kawahara, and Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
{komatani,kawahara,okuno}@i.kyoto-u.ac.jp

Abstract. For flexible interactions between a robot and humans, we address the issue of automatic recognition of human emotions during the interaction such as embarrassment, pleasure, and affinity. To construct classifiers of emotions, we used the dialogue data between a humanoid robot, Robovie, and children, which was collected with the WOZ (Wizard of Oz) method. Besides prosodic features extracted from a single utterance, characteristics specific to dialogues such as utterance intervals and differences with previous utterances were also used. We used the SVM (Support Vector Machine) as a classifier to recognize two temporary emotions such as embarrassment or pleasure, and the decision tree learning algorithm, C5.0, as a classifier to recognize persistent emotion, i.e. affinity. The accuracy of classification was 79% for embarrassment, 74% for pleasure, and 87% for affinity. The humanoid Robovie in which this emotion classification module was implemented demonstrated adaptive behaviors based on the emotions it recognized.

1 Introduction

A robot should be capable of interacting naturally with humans as a social partner and adapt its behavior according to his/her states. Emotions are important factors in reflecting these states [9], and therefore recognizing these plays an important role in dialogues particularly for entertainment. If a robot recognizes our emotions and responds in adaptation to these, we may feel social and friendly, which leads to more productive interaction.

Since speech interfaces play very important roles in human-robot interaction, automatic speech recognition (ASR) systems have recently been incorporated into robots for entertainment, such as pet and humanoid robots. Spoken dialogue technologies are also being introduced into them.

However, most recent research on spoken dialogue systems has only focused on verbal information contained in speech. Such systems, therefore, have tended to behave uniformly with all users when the verbal content of input sentences has been similar. Spoken dialogue, on the other hand, has many more characteristics than just verbal information. Such nonverbal characteristics also reflect individual user

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 413-423, 2004.

situations. The integration of nonverbal information should be taken into consideration to enable social interactions.

This paper focuses on emotional information, which has not been treated in conventional spoken dialogue systems. We present a method of automatically recognizing user's emotional states and achieving flexible dialogue based on emotions that can be recognized.

Most conventional studies into analyzing and recognizing speaker's emotions contained in speech have utilized prosodic features [2, 3, 8] and Kiebling et al. reported on these in detail [4]. We furthermore adopted another feature that is characteristics of dialogues, i.e. the interval between utterances. This is based on the assumption that this feature represents user’s embarrassment.

We also addressed the issue of the classification without prior learning because we wanted to apply the method to robots interacting with unknown visitors. In general, a user’s emotions included in speech are classified by comparing features in current utterances with those in his/her neutral states [7]. Therefore, data where a target user can be regarded as being in his/her neutral state is needed to normalize variations between individual users. We call the collection of data in their neutral states as prior learning. We designed several normalization methods that did not need prior learning, and attained comparable or better performance as a result.

There have been many classifications for human emotions such as anger, sadness, pleasure, calmness, surprise, and disgust. Huber et al. treated anger [2] and Lee et al. focused on negative emotions [6], to prevent customers on the telephone from hanging up. Our goal was to attain flexible interactions in a human-robot dialogue. We therefore focused on emotions that were important in spoken dialogue between humans and a robot, i.e., anger, pleasure, embarrassment, and affinity.

We evaluated our method using data collected from realistic situations. Many conventional studies have collected their data through having actors utter emotionally [7, 11]. We used data collected from children in a science museum with the WOZ (Wizard of Oz) method. The children's utterances were not pre-rehearsed but spontaneous. We also implemented our emotion recognition system in an interactive humanoid robot, Robovie [1], and achieved natural human-robot interactions.

2 Users’ Mental States in Dialogues with Robots

We focused on emotions that were important in smoothing interactions between robots and humans. These emotions were derived after analyzing corpora that had been obtained from children interacting with a robot using the WOZ (Wizard of Oz) method. We specifically handled the following four emotions.

- **Anger**
  Users are often hurt by speech recognition errors, which are unavoidable in speech communications. Utterances when users are angry make speech recognition even more difficult. By detecting this emotion, the system assumes there has been some misunderstanding, and generates a response to relieve this.

- **Pleasure**
  If users look pleased, it is assumed that they are enjoying themselves, and the system does not need to change the topic. It then listens further on the topic.
- **Embarrassment**
  If a user seems embarrassed about a topic, the system may change topics.
- **Affinity**
  There are many people who are not accustomed to talking with machines or robots. By detecting whether users are tense, the system can take action to alleviate this.

These emotions can be categorized into the following two according to their properties.

- **Temporary emotions**
  Temporary emotions vary per utterance, and affect the system's behavior during several utterances that follow. Anger, pleasure, and embarrassment can be categorized as temporary.
- **Persistent emotions**
  Persistent emotions depend on individual characteristics, and therefore do not change during one dialogue. The system's behavior is affected by the emotion throughout the whole dialogue. Affinity is categorized as persistent.

We then classified and evaluated temporary emotions per utterance and persistent emotions per speaker.

3 Target Data and Labeling

We used data that had been collected from visitors interacting with a robot using the WOZ method at the Kobe Science Museum. Most subjects were children aged from five to fifteen. The dialogue was in the form of questioning done by the robot on several topics. There were 46 dialogues, and a total of 498 utterances by subjects. The number of utterances ranged from ten and fifteen, and the duration of the dialogue lasted for several minutes. Figure 1 has a simple example dialogue.

Subjective labels were annotated for the data by three annotators, such as affinity (tense -- normal -- familiar), pleasure (pleased -- not pleased), and embarrassment (embarrassed -- not embarrassed). We did not treat anger because there was no data containing this emotion. This was because the data was collected with the WOZ

Table 1. Results of labeling for embarrassment and pleasure per utterance

Annotator	Embarrassment		Pleasure	
	Embarrassed	Not Embarrassed	Pleased	Not pleased
A	81	417	193	305
B	67	431	72	426
C	76	422	168	330
Agreement	36	405	47	251

method, and speech recognition errors had not occurred there. Annotators judged pleasure and embarrassment, which were temporary emotions, for every utterance. Affinity was determined for each dialogue. Table 1 lists the results of labeling for pleasure and embarrassment by the three annotators (A, B, and C in Table 1). Table 2 is for affinity, which was determined by a single annotator for the 46 dialogues.

R: Hi.
U: Hi.
R: I am Robovie. Who are you?
U: I am Natsuko.
R: Oh, your name is Natsuko. How old are you?
U: Seven.
R: Seven years old. Are you a second grader?
U: Yes. Ha... How do you know that?
(The subject is identified as pleased.)
R: Do you like your school?
(Asking the same topic further.)
U: Yes.
R: What is interesting at the museum today?
U: Well ... That is ...
(The subject is identified as embarrassed.)
R: OK, I will change the topic. Do you like me?
(Changing the topic.)
U: Yes, I do.
R: Good!!

Fig. 1. Example dialogue between subjects and robot by WOZ (R: robot, U: user)

Table 2. Results of labeling for affinity per dialogue

Affinity			
Familiar	Normal	Tense	Total
17	21	8	46

As we can see in Table 1, there were differences between the annotators. This means that impressions created by an utterance differ, and our aim was to recognize emotions that could be judged similar. We therefore used utterances for which the three annotators had given same labels in the experiment and evaluation that followed.

4 Automatic Classification of Users' Emotions

We now describe the automatic classification method of a user's emotions from his/her utterances. Although previous studies [7] needed prior learning where the characteristics of target users had been obtained beforehand, we used a method without the prior learning.

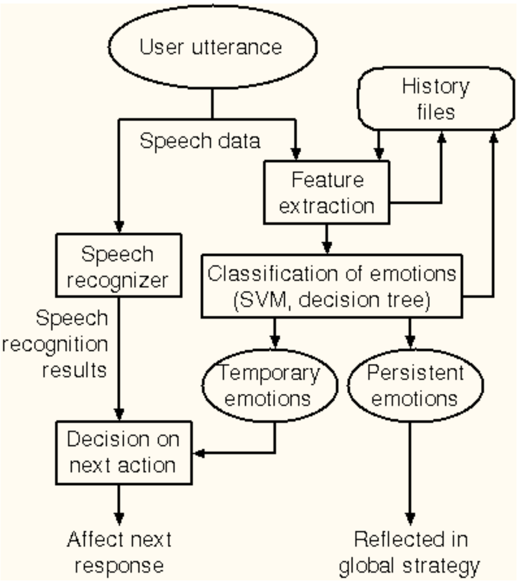


Fig. 2. Flowchart for system

Figure 2 is a flowchart for the system. It classifies a user's mental states from features contained in his/her speech. The classified mental states consist of temporary and persistent emotions as described in Section 2. User utterances are also transcribed by a speech recognizer. The next action is determined based on temporary emotions and the speech recognition results. The global strategy also changes based on the persistent emotion. Classification results and features of previous utterances are also taken into account as a history in addition to the features of the current utterance.

We used prosodic features that had been mentioned in many previous studies [2, 3, 4, 8], and the one we adopted can be listed as follows.

- Maximum F0 value
- Initial F0 value (onset)
- Average F0
- Difference between maximum and minimum F0
- Maximum power
- Average power
- Duration of utterance

Since we aimed at applying this method to robots interacting with unknown visitors, we could not normalize current utterances using values when users were calm. We therefore used another feature and normalization methods that were characteristic of dialogue, where the sequences of utterances were available. Specifically, we took the interval between the previous and current utterances, the difference of each feature with previous utterances, and normalization using the initial utterance of the dialogue into account.

- Difference between previous utterance and its normalization by current utterance
  The absolute values of features cannot be compared directly because they are individual. However, the differences between features are not affected by individuality, comparatively speaking. We considered both the differences and those normalized by features of the current utterance for each dimension.
- Normalization based on initial utterance
  We also normalized features of current utterances using those of initial utterances in the dialogue. This was based on the assumption that users at the beginning of the dialogue would be at their calmest.

We consequently adopted 29 feature values that consisted of the above seven features themselves and three operations (difference from previous utterance, difference normalized by current utterance, and normalization by initial utterance) for each of the seven features, and the interval between utterances. The interval was defined as the time between the end of the previous utterance and the beginning of the current utterance. We used the decision tree learning algorithm C5.0 [10] and the Support Vector Machine (SVM) [12] as classifiers. The linear kernel function was adopted for the SVM because there was not that much data.

5 Experiments and Evaluation

5.1 Experimental Conditions

We evaluated our method with the data described in Section 3, and the experiments were carried out with 10-fold cross validation. A process, where one tenth of all the data was used as the test data and the remainder was used as the training data, was repeated ten times, and the average accuracy was computed. We randomly changed the way the data was divided ten times, and computed the accuracy. The result obtained was averaged from a total of 100 calculations. The experiment for affinity was carried out by 5-fold cross validation because little data was available. To smooth out the unbalanced distribution of labels, we also introduced a cost corresponding to the reciprocal ratio of the number of samples in each class. This cost meant that the accuracy was computed under conditions where the number of samples was same for all classes.

5.2 Temporary Emotions (Embarrassment, Pleasure)

We will first describe the experimental evaluations we did on temporary emotions such as embarrassment and pleasure. As a baseline, we calculated the average values for utterances that were labeled as not having each emotion in the corpus, and normalized features of current utterances with this average. This baseline method corresponded to previous studies where features had been normalized by an utterance when users were calm that had been collected beforehand.
We calculated accuracy for the following conditions:

- Using both differences with the previous utterance and those normalized by the current utterance
- Using normalization based on the initial utterance
- Using the interval between utterances

We also calculated accuracy where all these 29 features were used.

Table 3. Classification accuracy for embarrassment and pleasure (decision tree)

Accuracy (%)	Embarrassment	Pleasure
Normalization by calm utterances (baseline)	63.1	66.9
Difference normalized by current utterances	59.3	66.3
Normalized by initial utterances	66.3	68.0
Using interval between utterances	66.4	68.8
Using all features	69.0	66.8

Table 3 lists classification accuracy made by the decision tree trained by the C5.0 [10]. Classification accuracy was 69.0% for embarrassment, which was better than with the baseline method, which is equivalent to doing prior learning. The interval between utterances often appeared in the higher parts of the decision tree for embarrassment. This meant the feature was effective in classifying embarrassment, and was independent of the other features as it improved accuracy by being used together with them. There were no dominant features for pleasure.

Table 4 lists the classification accuracy obtained with the SVM [12]. We attained an accuracy of 79.0% for embarrassment and 73.6% for pleasure, which exceeded those with the baseline method. We analyzed significant features in classifying

Table 4. Classification accuracy for embarrassment and pleasure (SVM)

Accuracy (%)	Embarrassment	Pleasure
Normalization by calm utterances (baseline)	73.5	71.8
Difference normalized by current utterances	78.3	73.6
Normalized by initial utterances	75.4	72.9
Using interval between utterances	76.6	72.5
Using all features	79.0	71.9

emotions by calculating accuracy where features were removed one by one. Features that played an important role in classifying embarrassment were maximum value of power, average F0, and intervals between utterances. The maximum value of power and its average were effective in classifying pleasure.

Table 5. Classification accuracy for affinity (C.5.0)

	Accuracy for three classes (%)	Accuracy for two classes (%)
Average for first utterance	44	66
Average for first two utterances	57	87
Average for first three utterances	56	79

Since our method obtained higher accuracy than the baseline, which needed prior learning, the features and operations we propose are appropriate in classifying emotions without prior learning.

5.3 Persistent Emotions (Affinity)

Let us now describe the experiment for affinity, which is a persistent emotion. Since a persistent emotion does not change greatly per utterance, we used the values for the seven prosodic features listed in Section 4 and the intervals between utterances.

A persistent emotion needs to be detected in the early stages of dialogue because it affects the global strategies that the system follows throughout the dialogue. We therefore calculated averages of the features for the first, first two, and first three utterances, and used them as features values in classification.

Classification was done with the decision tree (C5.0). We did not use the SVM because there was insufficient data for learning. The classification accuracy was calculated both for three classes (tense – normal – familiar) and for two classes: tense and others. This was because recognizing whether users were tense was more important in the dialogues.

Table 5 lists the classification accuracy for affinity. We attained an accuracy of 87% in classifying the two classes. The maximum value of power was effective in the classification.

6 Implementation in Robot

We installed the proposed classification module into a robot, Robovie, which was developed by ATR Intelligent Robotics and Communication Laboratories [1]. It was equipped with a number of sensors and movement mechanisms, and it communicated and interacted with humans through speech and gestures.

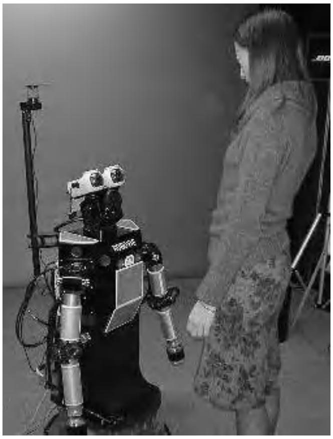


Fig. 3. Conversation with Robovie

R: Hello.
U: Hello.
R: Where are you from?
U: I come from Nagoya.
(Change next action through detected emotions)
R: Where is that?

[In case of pleasure]
R: Is it near or near? **(Pursue the topic)**
U: It is far from here.
R: You took a lot of trouble to come here,
didn't you?
R: I come from ATR.

[In case of embarrassment]
R: I come from ATR.
(Avoid pursuing the topic)

… …

R: Do you think I am pretty?
U: Yes, you are.
(Change next action by detected emotions)
R: I am very glad to hear that.

[In case of pleasure]
R: Which part of mine is pretty?
(Pursue the topic)
U: All.
R: All? Yeah!

[In case of embarrassment]
R: Let's shake hands?
(Change the topic)
U: O.K.

Fig. 4. Example dialogue with proposed models

We used Julian [5], which had been developed at our laboratory, for speech recognition. It decoded human speech into transcriptions using a specified grammar as a linguistic constraint. To reduce speech recognition errors, we restricted the vocabulary by preparing grammars and dictionaries for each state corresponding to system questions. The questions were deliberated to narrow down next user's responses. Stationary noise was removed by spectral subtraction.

Temporary emotions were classified by the SVM for every utterance, and persistent emotions were classified by decision trees that were trained by the C5.0 through the initial two utterances in the dialogues. Figure 3 is a photograph of a conversation between Robovie and a human subject. Figure 4 has an example dialogue.

7 Conclusion

We addressed the issue of flexible interactions between robots and humans, and investigated emotions such as embarrassment, pleasure, and affinity. The emotions were categorized into temporary emotions that changed per utterance and persistent emotions that did not change during dialogues. Conventional studies have needed prior learning to collect utterances in which users were calm. We, on the other hand, proposed the use of intervals between utterances and several operations that were specific to dialogue, such as calculating the differences between previous utterances and normalizing these using initial utterances. This enabled us to classify emotions without prior learning.

We also installed a classification module into a real robot. It changed its behavior according to emotions it recognized. Our future work will include evaluation of generated behaviors taking various experimental conditions and user characteristics into consideration.

Acknowledgements. The authors are grateful to Professor Hiroshi Ishiguro and Dr. Takayuki Kanda of ATR-IRC for their help in installing the emotion classification module into Robovie.

References

1. ATR Robovie. http://www.irc.atr.co.jp/~m-shiomi/Robovie/
2. Huber, R., Batliner, A., Buckow, J., Noth, E., Warnke, V., and Niemann, H.: Recognition of emotion in a realistic dialogue scenario. In *Proceedings of ICSLP* (2000), pp. 665-668.
3. Huber, R., Noth, E., Batliner, A., Warnke, V., and Niemann, H.: You BEEP Machine - Emotion in Automatic Speech Understanding System. In *Proceedings of TSD* (1998), pp. 223-228.
4. Kiebling, A., Kompe, R., Batliner, A., Niemann, H., and Noth, E.: Classification of Boundaries and Accents in Spontaneous Speech. In *Proceedings of the CRIM/FORWISS Workshop* (1996), pp. 104-113.
5. Lee, A., Kawahara, T., and Shikano, K.: Julius -- an open source real-time large vocabulary recognition engine. In *Proceedings of EUROSPEECH* (2001), pp. 1691-1694.
6. Lee, C. M., Narayanan, S. S., and Pieraccini, R.: Combining acoustic and language information for emotion recognition. In *Proceedings of ICSLP* (2002), pp. 873-876.
7. Moriyama, T., Saito, H., and Ozawa, S.: Evaluation of the Relationship between Emotional Concepts and Emotional Parameters on Speech. In *Proceedings of IEEE-ICASSP* (1997), Vol. 2, pp. 1431-1434.

8. Murray, I. R., and Amott, J. L.: Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *Journal of Acoustic Society of America*, Vol. 93, No. 2 (1993), pp. 1097-1108.
9. Picard, R. W.: Toward computers that recognize and respond to user emotion. *IBM Systems Journal*, Vol. 39, No. 3&4 (2000), pp. 705--719.
10. Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993. http://www.rulequest.com/see5-info.html
11. Schuller, B., Rigoll, G., and Lang, M.: Hidden Markov model-based speech emotion recognition. In *Proceedings of IEEE-ICASSP* (2003), Vol. 2, pp. 1-4.
12. Vapnik, V. N.: *Statistical Learning Theory*. John Wiley & Sons Inc., 1998.

Development of an Android Robot for Studying Human-Robot Interaction

Takashi Minato[1], Michihiro Shimada[1], Hiroshi Ishiguro[1], and Shoji Itakura[2]

[1]Graduate School of Engineering, Osaka University,
2-1 Yamada-oka, Suita, Osaka, 565-0871, Japan
{minato,ishiguro}@ams.eng.osaka-u.ac.jp
shimada@ed.ams.eng.osaka-u.ac.jp

[2]Graduate School of Letters, Kyoto University,
Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
itakura@psy.bun.kyoto-u.ac.jp

Abstract. Behavior or Appearance? This is fundamental problem in robot development. Namely, not only the behavior but also the appearance of a robot influences human-robot interaction. There is, however, no research approach to tackling this problem. In order to state the problem, we have developed an android robot that has similar appearance as humans and several actuators generating micro behaviors. This paper proposes a new research direction based on the android robot.

1 Introduction

In recent years, there has been much research and development of intelligent partner robots that can interact with humans in daily life, such as Sony AIBO and Honda ASIMO. In this research, communication between the robots and humans is emphasized in contrast to industrial robots performing specialized tasks. Meanwhile, the intelligence of a robot is a subjective phenomenon that emerges during human-robot interaction. It is, therefore, indispensable to reveal a principle of human-robot and human-human communication, that is, a principle of interaction for developing a partner robot and realizing its intelligence.

Some researchers have tackled this problem. For example, Kanda et al. [1] and Scheeff et al. [2] evaluated how the behavior of their robots affects human-robot interaction by observing their interaction. These works have gradually revealed the effects of robot behavior on human-robot interaction. There is, however, a possibility that robotic appearance distorts our interpretation of its behavior. The appearance of the robot is essentially one of its functions; therefore, the effect of appearance must be evaluated independently. It is generally difficult to isolate the effects of a robot's behavior from those of the robot's appearance which is dissimilar from humans. One way to discriminate is developing a robot whose appearance is the same as humans.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 424-434, 2004.

This paper proposes a new research direction to tackle a fundamental problem of a robot behavior and appearance using a robot called an *android* which has similar appearance to humans. To state the problem, we form a fundamental hypothesis about the effect of a robot's behavior and appearance using existing knowledge gained from research or practical experience. Moreover, we design experiments using the developed android. This research is currently in progress and only preliminary experimental results have been obtained. In this paper we describe the hypotheses and the developed android, and finally present brief results of preliminary experiments.

The rest of paper is organized as follows. Section 2 and 3 describe our research objective and hypotheses. Section 4 introduces the android robot developed for this research. Section 5 shows brief results of the experiments to observe behavior of subjects while interacting with the android. Finally, Section 6 presents conclusions.

2 Purpose and Approach

Our goal is to create a design methodology of robot behavior and appearance to realize natural communication between robots and humans. Our approach is to form a hypothesis about the effects of behavior and appearance on interaction and examine the hypothesis in a psychological experiment.

Kanda et al. [1] investigated the effects of interactive behavior using a humanoid robot named "*Robovie*" [3]. It is, however, possible that the results depend on the appearance of Robovie because its robotic appearance influences the interaction. In the psychological field, Johnson et al. [4] reported that infants followed the gaze of a novel object that had facial features and contingent interactivity and did not follow an object that did not have facial features or contingent interactivity. According to this evidence, it is clear that the effect of a robot's appearance cannot be ignored.

There is a bottom-up approach to tackle the "behavior versus appearance problem," in which the interaction is evaluated while incrementally enhancing the behavior or appearance of the robot. However, there is also a top-down approach, in which we initially build a robot which has the same motion and appearance as humans and evaluate the interaction while removing some aspect of behavior or appearance.

To employ the later approach, we introduce an android robot that has a similar appearance as humans. McBreen and Jack [5] evaluated some human-like agents which were created from human photorealistic images in an e-retail application (a home furnishings service). The results show that the conversation with the video agent is thought to be more natural than the conversation with the other agent (e.g., a 3-D talking head, a still image with facial expressions, and a still image). This work suggests that the close resemblance to humans removes the effect of the robot‘s dissimilar appearance and enables an investigation purely of the effect of behavior. Comparing the results with the android and other humanoid robots, the effects of behavior and appearance are extracted independently.

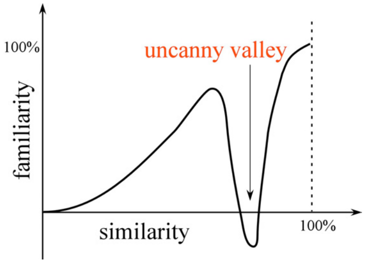


Fig. 1. Uncanny valley

In traditional robot research, the design of a robot appearance has been entrusted to an artistic designer and not had an engineering meaning. However, the robot's appearance can be designed based on the engineering methodology from our result.

In this research, it is necessary to evaluate human behavior in the interaction with the android. There are qualitative and quantitative methods to evaluate. Kanda et al. [6] employed a qualitative method by measuring the psychological attitudes of people using the semantic differential method (SD). However, it is difficult to prepare opposite pairs of adjectives in a questionnaire to obtain a result that is explicable.

Some researchers quantitatively evaluate human behaviors. For example, Matsuda et al. [7] investigated the brain activities of people who were playing a video game using Near-Infrared Spectroscopy (NIR). Kanda et al. [8] quantitatively evaluated behaviors of people who were in communication with Robovie using the motion capture system and eye mark recorder. According to these studies, we employ the quantitative method using a motion capture system and eye mark recorder.

3 Hypotheses about Appearance and Behavior

Mori [9] mentioned the relationship between familiarity and similarity of robot appearance and motion to humans. Familiarity of a robot increases with its similarity of appearance and motion until a certain point, when a subtle imperfection of the appearance and motion becomes repulsive (Fig. 1). This sudden drop is called an "uncanny valley." In the figure, appearance and motion are evaluated on the same axis. It is, however, not always the case that they are evaluated in the same manner.

We hypothesize that robot's appearance and behavior independently influence human-robot interaction. Namely, an identical behavior can differently influence if the appearances are different. With respect to robot's appearance, our hypothesis is the following:

- The evaluation of interaction increases with similarity of robot's appearance. At the point of closely resemblance to humans, there is a valley like "uncanny valley" as shown in Fig. 2 (a). The depth of the valley decreases with complexity of robot's behavior.

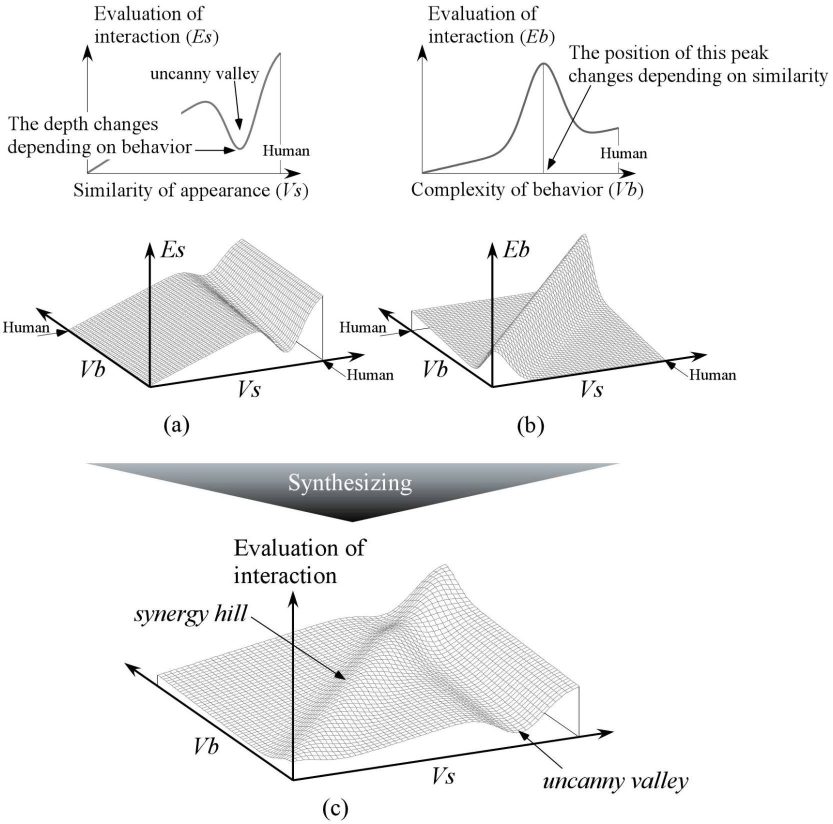


Fig. 2. There are two factors that influence interaction: behavior and appearance. (a) An evaluation of interaction plotted against similarity of appearance. There is uncanny valley. (b) An evaluation plotted against complexity of behavior. There is a peak means synergy effect of appearance and behavior. (c) Synthesized evaluation. There are two features: "uncanny valley" and "synergy hill." Actually, each variable cannot be represented in one axis.

Goetz et al. [10] proposed the "*matching hypothesis*" that the appearance and social behavior of a robot should match the seriousness of the task and situation and examined it in a psychological experiment with the Nursebot robot, "*Pearl*." The result suggests that human-like behavior does not always make a good impression and that the robot's appearance determines what behavior is appropriate. We hypothesize that there is a synergy effect of a robot's appearance and behavior.

- The evaluation increases with the complexity of the robot's behavior. At the point of matching robot's appearance, there is a synergy effect of appearance and behavior shown as a peak in Fig. 2 (b).

Synthesizing two hypotheses, the evaluation of interaction is qualitatively represented as Fig. 2 (c). The robot's uncanny appearance is mitigated by its behavior if the behavior closely resembles that of humans.

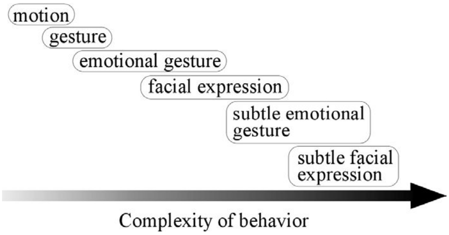


Fig. 3. Hypothesis about complexity of behavior. Subtle emotional behaviors including facial expressions are human-like behaviors.

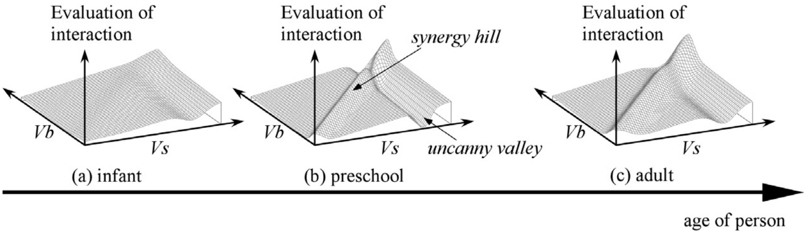


Fig. 4. Hypothesis about person's age.

Many factors influence the complexity of behavior. One of them is emotion. In general, one feels frustration in communicating with a person who keeps a straight face. It seems uncontroversial to assume that emotional behavior including facial expressions is human-like behavior. Fig. 3 shows the hypothesis about the complexity of behavior. A simple motion empty of meaning is less complexity, and a subtle emotional gesture and facial expression have high complexity and similar to human behavior.

In the above, we focus on the attributes of a robot. The attributes of a person (e.g., age and gender) interacting with a robot influence the interaction. To compare a subject's reaction at different ages, a couple of infants less than 13-months old and preschool children from three to five years old directed toward the developed android. As a result, infants seemed to be attracted by the android. However, children were afraid of the android at a glance and unwilling to face it. The behavior of children is explained in terms of Mori's "uncanny valley." The result suggests that the uncanny valley seems to change owing to person's age. With respect to person's age, we hypothesize as follows:

- The uncanny valley becomes the deepest in early childhood and shallower in adulthood.
- A synergy hill (see Fig. 2) becomes the steepest at younger children and smoother at adults.

Fig. 4 illustrates the hypothesis. We will next form hypothesis about other attributes.

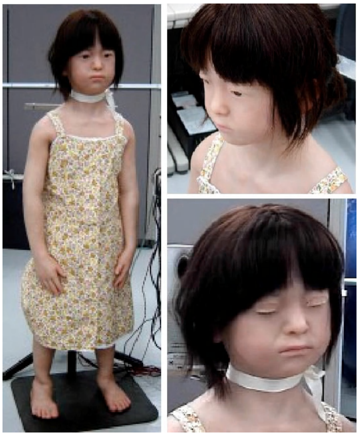


Fig. 5. The developed android robot named "Repliee R1." Left: External appearance. Upper Right: Head appearance. Lower right: Head appearance with eyes closed.

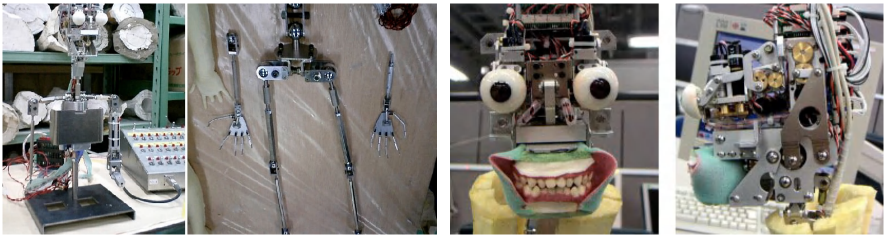


Fig. 6. The skeleton of the android.

4 The Developed Android Robot

Fig. 5 shows the android robot named "*Repliee R1*" that is developed as a prototype. To make the appearance closely resemble humans, we made a mold of a girl, and we carefully chose a kind of silicon that would make the skin feel human-like. The appearance is a five-year-old Japanese girl. The prototype has nine DOFs in the head (five for the eyes, one for the mouth and three for the neck) and many free joints to make a posture. The actuators (motors) are all embedded inside the body.

The touch sensor used in the android is a strain rate force sensor. The mechanism is similar to human touch insofar as it detects touch strength while the skin is deforming. The android has four touch sensors under the skin of the left arm (Fig. 7). Only four sensors can measure the touch strength all over the surface of the left arm. These tactile sensors enable various touch communications.

The android shown above is developed as a prototype. In the future, we will implement an android with the same number of joints as humans, tactile sensors covering the whole body, vision sensors, and auditory sensors.

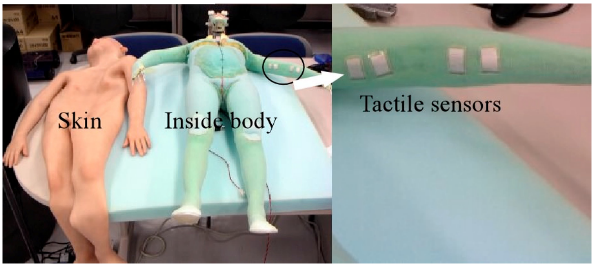


Fig. 7. Skin, inside body, and tactile sensors. Space between skin and skeleton is filled with urethane foam, which can be replaced with other mechanism.

5 Preliminary Experiment

5.1 Study of Gaze Behavior

For a quantitative evaluation of interaction, we investigated eye motion of people during a conversation with the android. An evaluation of semi-unconscious behavior such as eye motion can reveal facts that do not appear in qualitative evaluations, such as a questionnaire test. We predicted that gaze behavior would vary owing to the similarity of a robot's appearance and the complexity of its behavior during communication. To test the prediction, three types of actors (interlocutors) were prepared: (*A1*) a human girl, (*A2*) the android with eye, mouth, and neck motions, (*A3*) the still android. The girl was a five-year-old Japanese girl and was not shy with strangers. Subjects were 18 Japanese undergraduate and graduate students. There were 10 males and 8 females. A subject had a brief conversation with each actor in random order with replacements except excluding. To control the conversation, we designed the following script.

Conversation Script (an English translation)
Actor: Hi, I'm [name].
Subject: [answers]
Actor: Let's play together! I'll give you a quiz. Are you ready?
Actor: What is a word starting with [any alphabetic character]?
Subject: [answers]
Actor: That's right! Well, what is a word starting with [any alphabetic character]?
Subject: [answers]
Actor: No! Well, then, what is a word starting with [any alphabetic character]?
Subject: [answers]
Actor: That's right! That was fun! Bye-bye!

This is only a sample script. The order of the robot's positive and negative responses may differ. The conversation was held in a small room partitioned by a curtain (Fig. 8.) The experimenter behind the curtain controlled reactions of the android. A speaker produced the prerecorded voice of the android. *A2* moved its mouth while

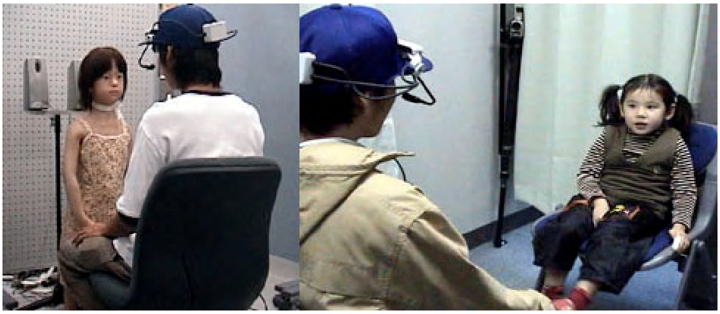


Fig. 8. Experimental room. A subject mounting an eye mark recorder has a brief conversation with the android (Left) and the girl (Right.)

talking and sometimes blinked and moved its neck, but *A3* was stationary even when it was talking.

An eye mark recorder (NAC EMR-8) measured the eye motion with the rate of 30 Hz. We defined a gaze fixation as a gaze fixed for more than four frames (133 msec) and counted the frequency that the subject fixated on the actor's eyes (including glabella), nose and mouth in each conversation. At the end of the experiment, the subject answered an open questionnaire about his or her impression of the actor.

5.2 Results

Fifty-four conversations (18 subjects × 3 actors) in total were held. Ten data were omitted owing to much detection error. Table 1 shows the mean frequencies of the subjects' fixation falling on the actors' eyes, nose, and mouth. A one way ANOVA showed that there was a significant difference (F =3.32, $p < 0.05$) between actors with respect to the frequency of fixation falling on the eyes and no significant difference with respect to the nose and mouth. Furthermore, a t-test showed that there were significant differences between *A1* and *A2* ($t = 3.10$, $p < 0.005$) and *A1* and *A3* ($t = 2.45$, $p < 0.05$) with respect to eyes. Fig. 9 shows the distributions of fixation points that fell on the face of the android and girl. Brighter points indicate high frequency of fixation.

The result shows that subjects look at the android's eyes more frequently than girl's, although Japanese people tend to avoid eye contact owing to cultural reasons [11]. The subject's mental state may explain differences in gaze behavior [12, 13]. One possibility is to assume that the subjects tried to achieve mutual understanding. Many subjects felt artificiality of the android's eye movement rather than mouth

Table 1. Mean frequencies of fixation per second (standard deviations in parentheses).

	A1	A2	A3
Eyes	0.30 (0.059)	0.92 (0.57)	0.82 (0.52)
Nose	0.085 (0.013)	0.15 (0.016)	0.13 (0.016)
Mouth	0.0014 (3.2×10^{-6})	0.0029 (1.3×10^{-5})	0.0017 (4.7×10^{-6})

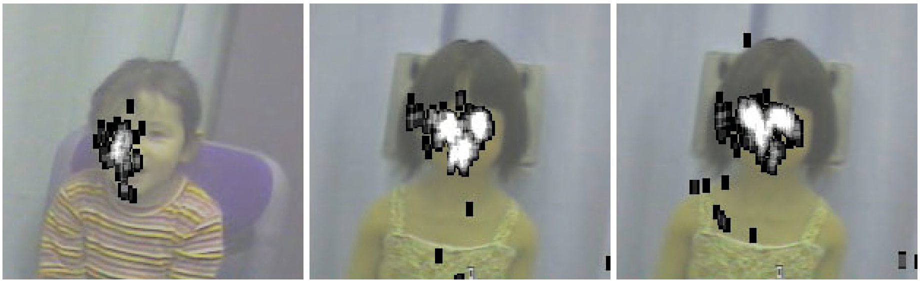


Fig. 9. Distribution of fixation point fell on the girl (Left), android A2 (Middle) and android A3 (Right). Brighter point means high frequency of fixation.

movement. It was found from the result that how subject's gaze at the android, especially at its eyes, differs from that at humans. This result is important, because it is possible that the difference in the effect of the robot's appearance and behavior on human-robot communication is evaluated by measuring the participant's gaze point as well as the subjects' gaze point in human-human communication. A human gaze is a semi-unconscious behavior and reflects a hidden factor which cannot be self-reported. It is expected that measuring gaze behavior would find an effect of a robot's appearance and behavior that would not appear in the answers to a questionnaire.

5.3 Discussion

We predicted that there was a difference in the subjects' gaze behavior between *A2* and *A3*. Contrary to our prediction, there was no significant difference. The result showed that the random eye lids and neck motion and mouth motion synchronized with voice did not influence the conversation with the android. It is considered that the experiment was lacking in some assumptions. This section discusses further hypotheses about the android's appearance and behavior from the observations of gaze behavior and answers to the questionnaire.

Uncanny valley
Many subjects mentioned that artificiality of the android's appearance, behavior and imbalance between appearance and behavior on the questionnaire. The artificiality of eye motion in particular may cause an increase in the number of fixations on the android's eyes. Furthermore, the high frequency of fixation could represent the uncanny valley shown in Fig. 2. To examine this prediction, it is necessary to ascertain whether subjects provide fewer fixations on a robot that has robotic appearance, such as ASIMO. We hypothesize that the frequency of fixation represents the evaluation of communication, and the evaluation varies inversely with the frequency.

Eye contact
Some subjects mentioned that they could not make eye contact with the android. It is considered that the lack of eye contact causes the uncanniness. Some psychological

researchers show that eye contact can serve a variety of functions (e.g., [12, 13]) in human-human communication. It is estimated that eye contact and the android's appearance work synergistically to enhance communication. To ascertain this, we will compare with a robot that has a robotic appearance and no eye contact behavior.

Contingent motion
One subject answered that the android with motion (*A2*) was more uncanny than the still android (*A3*) because the motion was not contingent. Another subject mentioned that repeating same behavior of the android was unnatural. It is possible that the lack of the contingent android's motion (*A2*) made no difference between *A2* and *A3* in the result. As described in section 2, a contingent motion of nonhuman object varies an infant's attitude [4]. It is estimated that a contingent motion of the android provides an effect that works in synergy with its human-like appearance.

Involuntary waving motion
One subject mentioned that it was uncanny that the android (*A2*) was moving only the head though human interlocutor (*A1*) was always moving the whole body slightly. Miyashita and Ishiguro [14] showed that the slight involuntary waving motion of a humanoid robot makes its behavior more natural. It is quite likely that a slight involuntary waving motion of the whole body seems animate living. To state that the involuntary motion provides a synergy effect, however, it is necessary to compare the android and other robots.

Habituation effect
All the subjects in the experiment were only those who saw the android for the first time. In other words, they were not familiar with the android yet; therefore, the habituation effect cannot be ignored. Some subject answered that they were surprised at the android in the first conversation but familiar with it in the second conversation. All of their gaze behavior showed that the frequency with which fixation fell on the android's eyes in the second conversation decreased from that of the first conversation. Habituation to the android seems to change the interaction. In section 3, we hypothesized that person's age changes the human-robot interaction. We must, however, investigate the short-term (order of minutes or hours) change of interaction.

6 Conclusion

This paper has proposed a new research direction based on the android robot to reveal a principle of human-robot interaction. An evaluation of this interaction is impeded by the difficulty of isolating the effect of behavior from that of appearance. The appearance of the android, however, may decrease the effect of robot appearance. Furthermore, this research gives a methodology for robot design, which had previously been entrusted to an artistic designer.

This paper has shown the fundamental hypotheses about the effects of robot behavior and appearance on human-robot interaction and the preliminary experiments to observe human reactions to the android. We are still in the progress of forming more detailed hypotheses and designing experiments.

References

1. Kanda, T., Ishiguro, H., Ono, T., Imai, M., Mase, K.: Development and Evaluation of an Interactive Robot "Robovie", *IEEE International Conference on Robotics and Automation* (2002) 1848-1855
2. Scheeff, M., Pinto, J., Rahardja, K., Snibbe, S., Tow, R.: Experiences with Sparky, a Social Robot. *Workshop on Interactive Robot Entertainment* (2000)
3. Ishiguro, H., Ono, T., Imai, M., Kanda, T., Nakatsu, R.: Robovie: an Interactive Humanoid Robot. *International Journal of Industrial Robot*, Vol. 28, No. 6 (2001) 498-503
4. Johnson, S.C., Slaughter, V., Carey, S.: Whose Gaze Will Infants Follow? Features that Elicit Gaze Following in 12-month-olds, *Developmental Science*, Vol. 1 (1998) 233-238
5. McBreen, H.M., Jack, M.A.: Evaluating Humanoid Synthetic Agents in E-Retail Applications, *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans,* Vol. 31, No. 5 (2001) 394-405
6. Kanda, T., Ishiguro, Ishida, T.: Psychological Analysis on Human-Robot Interaction, *IEEE International Conference on Robotics and Automation* (2001) 4166-4173
7. Matsuda, G., Hiraki, K.: Frontal Deactivation in Video Game Players, *Annual Conference of International Simulation and Gaming Association* (2003) 799-808
8. Kanda, T., Ishiguro, H., Imai, M., Ono, T.: Body Movement Analysis of Human-Robot Interaction, *International Joint Conference on Artificial Intelligence* (2003) 177-182
9. Mori, M.: *The Buddha in the Robot*, Charles E. Tuttle Co., Japan (1982)
10. Goetz, J., Kiesler, S., Powers, A.: Matching Robot Appearance and Behavior to Tasks to Improve Human-Robot Cooperation, *IEEE Workshop on Robot and Human Interactive Communication (2003)*
11. Axtell, R.E. (Ed.): *Do's and Taboos Around the World*, John Wiley & Sons, New York (1993)
12. Argyle, M.: *The Psychology of Interpersonal Behaviour*, Penguin Books, London (1967)
13. Argyle, M., Cook, M.: *Gaze and Mutual Gaze*, Cambridge University Press, London (1976)
14. Miyashita, T., Ishiguro, H.: Natural Behavior Generation for Humanoid Robots, *IEEE International Conference on Humanoid Robots* (2003)

Open-End Human Robot Interaction from the Dynamical Systems Perspective: Mutual Adaptation and Incremental Learning

Tetsuya Ogata[1,2,3], Shigeki Sugano[2], and Jun Tani[3]

[1] Graduate School of Informatics, Kyoto University, Yoshida-honmachi Sakyo-ku,
606-8501 Kyoto, Japan
ogata@i.kyoto-u.ac.jp

[2] Humanoid Robotics Institute, Waseda University, 3-4-1 Okubo Shinjuku-ku,
169-8555 Tokyo, Japan
{ogata, sugano}@paradise.mech.waseda.ac.jp

[3] Brain Science Institute, RIKEN, 2-1 Hirosawa Wako-shi,
351-0198 Saitama, Japan
{ogata, tani}@brain.riken.jp

Abstract. This paper describes interactive learning between human subjects and robot using the dynamical systems approach. Our research concentrated on the navigation system of a humanoid robot and human subjects whose eyes were covered. We used the recurrent neural network (RNN) for the robot control. We used a "consolidation-learning algorithm" as a model of hippocampus in brain. In this method, the RNN was trained by both a new data and the rehearsal outputs of the RNN, not to damage the contents of current memory. The proposed method enabled the robot to improve the performance even when learning continued for a long time (open-end). The dynamical systems analysis of RNNs supports these differences.

1 Introduction

Many kinds of the mechanical systems that cooperate with human beings have recently been studied in efforts to improve task performance and the mental impression of the persons using the system. A humanoid robot, for example, will not only have to help people work but also have to establish a new relationship with people in daily life.

We focused on interactive learning between a human operator and a robot system, in a fundamental form to design natural human-robot collaboration. It consists of the robot system, which learns the task including a human operator, and the human, who learns the task including the robot system. It is usually difficult to stabilize the system for a long period of time of operation because the *mutual adaptation* in such coupled and nested systems between humans and robots tends to generate quite complex dynamics. One of the difficult and interest an issue of such an open-end interaction is an

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 435-444, 2004.

incremental learning because robots have to continue to adapt humans who change motion strategies every time. However, the incremental learning of machine is generally difficult especially when there are contradictions between a new learning data and current memories.

Although there have already been some studies of learning systems in man-machine cooperation [1][2], most of them only focused on short period operations in which the cooperation relation between the person and the machine is organized. Therefore, they did not discuss important aspects such as the mutual interaction after the relation organization, the collapse and modification of the relation, and the long process of development from a beginner to an expert.

Miwa et al. performed an experimental study [3] exploring the collapse and modification of relationships between people, but such phenomena are hard to analyze because human learning and cognitive processes cannot be measured directly. Miyake et al. studied the walking cooperation between a person and a robot model [4], but because of their simple modeling using a nonlinear oscillator, their analysis was limited to some simple phenomena such as the synchronization and revision of the walking rhythm.

To investigate interactive learning for a long time, we developed a navigation task performed by a humanoid robot. This paper describes the results of our experiments and the validity of the "consolidation learning" method implemented to ensure the robustness of neural network output.

2 Navigation Task

A navigation task is employed in which a humanoid robot, Robovie, developed in ATR [5], and a human subject navigate together in a given workspace. The height of Robovie is 1200 mm and the weight is about 60 kg. It has various features enabling it to interact with human beings: two arms with four degrees of freedom, a humanlike head with audiovisual sensors, and many tactile sensors attached to its body. Photographs of Robovie and the navigation task are shown in Fig. 1. The experimental environment used was a 4x4-m L shaped course of which outside walls were marked red and blue for every block. Robovie and the human subject held their arms together and attempted to travel clockwise in the workspace as quickly as possible without hitting obstacles. The actual movement of the robot and the subject is determined by adding two motor forces; one is the motor vector determined from a neural network in the robot and the other is the subject's directional control force exerted to the robot's arms. The neural network in the robot is adapted incrementally after each trial of travel based on the travel performance. The performance is measured by the travel time period at each trial.

An interesting point of this collaboration task is that the sensory information is quite limited for both the robot and the subject. The robot can access only local sensory information such as ultrasonic sensors and a poor vision system (it only detects vague color information of its surroundings), but not for exact global position information. The subject's eyes are covered during the navigation task, however the subject is allowed to look around the workspace before the experiments begin. The subject has to guess his/her situation or position by means of the interactive force felt between the robot and his/her arms utilizing his/her memorized image of the workspace geometry. Both sides attempt to acquire the forward model of anticipating the future sensory image as well as the inverse model of generating the next motor commands utilizing the poor sensory information of different modalities from past experiences.

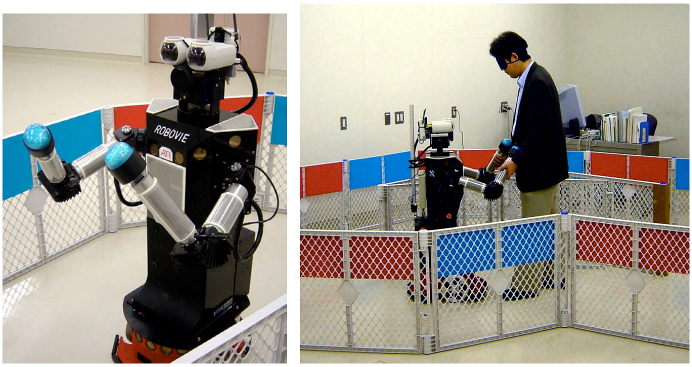


Fig. 1. Humanoid Robot, Robovie and Navigation Task

3 The Model and System

3.1 Neural Network Architecture

In many cases the actual states of the systems cannot be identified explicitly just by looking at the current sensory cues, but they do through more context-dependent manners by utilizing the history of the sensory-motor experiences. In our experiment case, the current sensory inputs may not tell the exact position of the robot due to the sensory aliasing problems. This is called the hidden state problem. Long-Ji Lin as well as Jun Tani have shown that the recurrent neural network (RNN) can solve this problem where the so-called context units are self-organized to store the contextual information [6, 7]. We applied the Jordan type recurrent neural network (RNN) [8] in which context units are added to the usual-feed forward neural network (FFNN).

Fig. 2 shows the RNN architecture design of the robot. The RNN operated in a discrete time manner with the synchronizing of each event, and the input layer of the RNN consisted of the current sensory input and the current motor values. The sensory inputs are comprised of the output of the ultrasonic range-sensors and the color area acquired from the omni-direction camera mounted on the robot's back. The motors consist of the current forward velocity and rotation velocity. The input layer has only seven units. The output layer also has seven units, and its outputs are the prediction of the next sensory input and the next action. This is the implementation of the paired forward and inverse model proposed by Kawato [9]. There are forty context units in the input and output layers. The activations of the context outputs in the current time step is copied to those of the context inputs in the next time step. It is noted that the context units activities are self-organized through learning processes such that they can represent the current state of the system corresponding to the past input sequences.

In our application, the RNN is utilized not only as a mapping function from inputs to outputs but also as an autonomous dynamical system. Concretely, the RNN can have two modes of operations as shown in Fig. 3. The first mode is the open-loop mode where one-step prediction of the sensory-motor prediction is made using the inputs of the current sensory-motor values. The second mode is the close-loop mode in which the output re-entered the input layer through the feed back connection. By iterating this with the closed loop, the RNN can generate an arbitrary length of the look-ahead prediction for future sequences with given initial states in the input layer. This function for the look-ahead prediction of the sensory-motor sequences can achieve the mental rehearsal [10] that will be described later in the explanations of the consolidation learning. The middle layer had thirty neurons. The RNN was trained by using the back propagation through the time (BPTT) learning method [11].

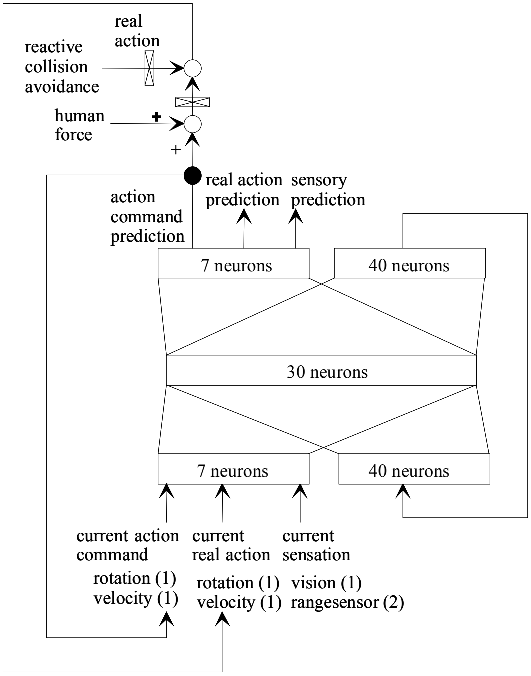


Fig. 2. Neural Network Architecture

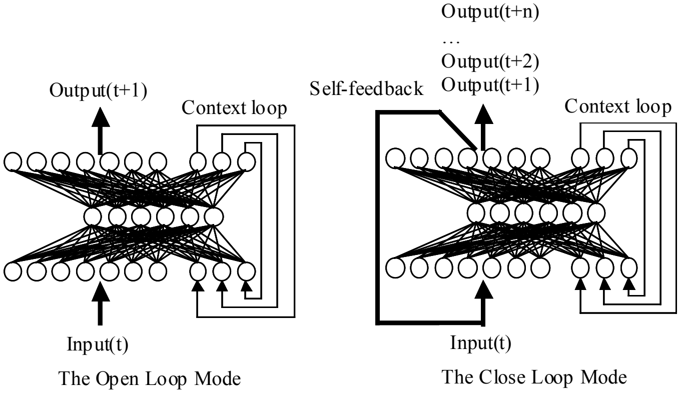


Fig. 3. Open Loop Mode and Close Loop Mode

3.2 Consolidation Learning

It is generally observed that if the RNN attempts to learn a new sequence, the contents of the current memory are severely damaged. One way to avoid this problem is to save all the past teaching data in a database, add new data, and use all the data to retrain the network. The problem with this method, however, is that the learning time of the RNN is increased by increasing the amount of stored data.

To solve this problem, we used new learning method for RNN proposed by Tani [10]. Observations in biology show that some animals use the hippocampus for temporary storage of episodic memory and consolidate them into neocortical systems as long-term memory during sleep. Tani modeled this process by using an RNN and a database. In this method the newly obtained sequence pattern is stored in the "hippocampus" database. The RNN, which corresponds to the neocortex, rehearses the memory patterns, and these patterns are also saved in the database. The rehearsal can be performed in the close-loop mode described in the previous section. Various sequence patterns can be generated by setting the initial state of the RNN differently. The ensembles of such various rehearsed sequences actually represent the structure of the past memory in the dynamical systems sense. The RNN is trained using both the rehearsed sequential patterns which correspond to the former memory and the current sequence of the new experience. We named this method "consolidation-learning algorithm".

It is expected that this algorithm enable the RNN to carry out incremental learning while maintaining the structure as much as possible. Although some robot studies using this algorithm have been performed, its detailed characteristics have not yet been clarified.

3.3 Navigation System

Since the robot moves using the RNN, its performance is inadequate in the initial stage of learning. We therefore implemented a collision avoidance system which overrides the RNN commands when the minimum output of a range sensor falls below the threshold value. This system is just the reflection system tuned up by the designer. In our experiments, the more overrides made by this man-made collision avoidance system mean the less performance of the RNN. The robot obtained the color area, range sensor data, and vehicle conditions every 0.1 s. This data was compressed and filtered. The RNN receives this preprocessed data as input and generates the output with a time interval of 2 seconds.

A simplified reinforcement-learning method was employed for the RNN learning as follows. At each trial, the robot and the subject go around the workspace together for a fixed number of times. Then the time period taken for this travel is measured. If the performance in terms of the time period is better (less period) than the previous trial, the RNN is trained with the sensory-motor sequence experienced with this trial (with rehearsed ones in the consolidation learning). Otherwise, no training is conducted on the RNN. At each trial, 3,000 steps of iterative learning is conducted offline using the external computer. In this way, the learning in the robot side is conducted incrementally depending on the performance achieved at each trial.

4 Experiments

The learning algorithms were evaluated and compared in 15-trial navigation experiments with seven male subjects. In each trial, the subject and the robot went around the workspace two times. After each trial there was a one-minute break for the questionnaire, which consisted of 11 items [1]. Additionally, at the end of each experiment, the subjects filled out the questionnaire based on NASA-TLX [12].

Three neural networks, the FFNN, the RNN with a usual learning method, and the RNN with consolidation learning explained in Section 3.2 were compared in the experiments involving the seven subjects. In consolidation learning, the teaching data consisted of the current sequence pattern and the three rehearsal patterns. These three rehearsal patterns were generated in the close-loop mode with changes in the initial value of the context units randomly, and stored in the database outside the RNN. The order of the experiments was changed with the subjects to avoid presenting subjects with a fixed order that might influence the results.

As the result of the experiments, although all performances improved in the first half of the learning, differences appeared in the second half. The performance of the FFNN gradually deteriorated and that of the RNN with the usual learning method stagnated as the subject changed the operation by learning on its own. Only the performance of the RNN with the consolidation learning continued to improve.

The results of the NASA-TLX questionnaire and the 11-item questionnaire are shown in Fig. 4. In each questionnaire, the significant values of the levels of 1 % and

5 % were calculated by Scheffe-test. It is easy to see that in both questionnaires the RNN with the consolidation-learning algorithm gave the best mental impressions.

It is interesting here to compare the results between the FFNN and the RNN with the usual learning method. In the robot experiments, the performance of the RNN was better than that of the FFNN. In the navigation experiments, however, the FFNN tended to give the subjects a better impression than the RNN especially in "result of work", "fatigue free" and "operability" in the 11-item questionnaire.

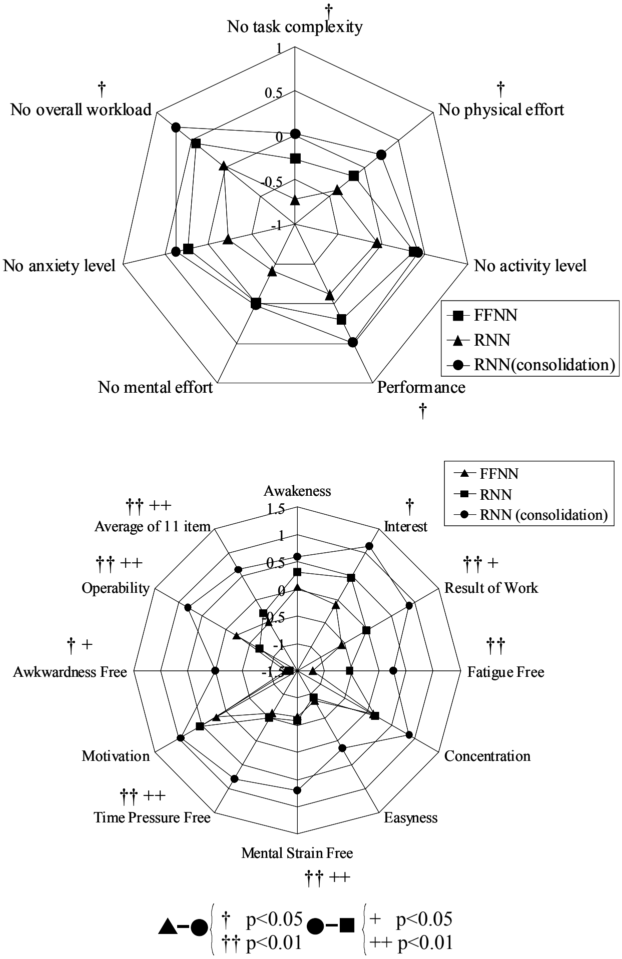


Fig. 4. Results of Questionnaire of NASA-TLX and 11-item Evaluation

5 Discussion

In the experiment with only the robot, the RNN performed effectively, because the robot could decide the action using not only the sensor input including the noises but also the information of the context layer. In the human-robot cooperation, however, the performance of the RNN with the usual learning was worse than that of the FFNN. It is thought that this is due to interactive learning including "Incremental learning" which damaged the memory of the RNN. As the result, the "operability" became worse because the robustness of the RNN to the input noise decreased.

To analyze the effect of consolidation learning, we examined the robustness of the RNN dynamics by looking at its initial sensitivity characteristics. Both RNNs obtained after the usual learning and the consolidation learning in our experiments were tested to generate the output sequences in the close-loop mode with the addition of three different sizes of noise in the initial input values. Fig. 5 and 6 shows the motion trajectories of the robot re-constructed from the rehearsal motor output of the RNN. These represent how the output trajectories developed with small differences in the initial input conditions for the RNNs with usual learning and consolidation learning. It is observed that output trajectories of the RNN by the usual learning tend to diverge more than those by consolidation learning. This implies that the usual learning scheme tends to generate more unstable dynamic structures in the trained RNN.

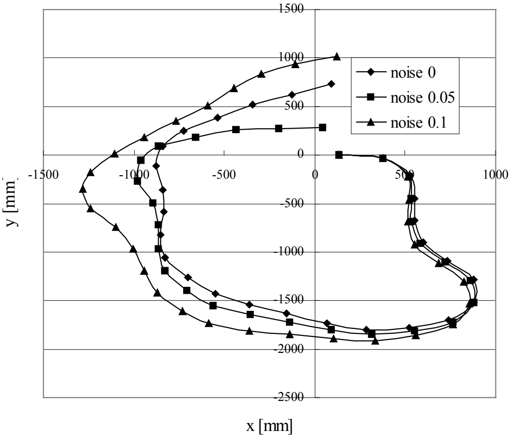


Fig. 5. Examples of Rehearsed Trajectory with Input Noises (The RNN with usual-learning method)

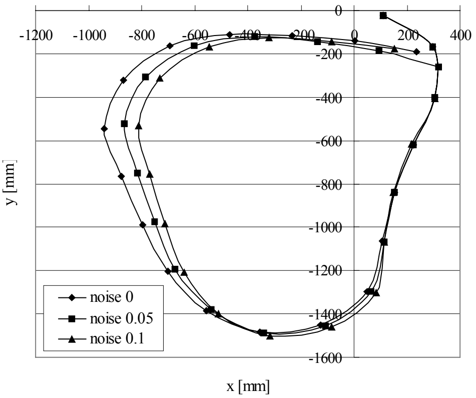


Fig. 6. Examples of Rehearsed Trajectory with Input Noises (The RNN with consolidation-learning method)

This robustness characteristic of the RNN seems to be directly related to the "operability" in the mental impressions. If the RNN tends to diverge largely even with small deviations in the input sequences from the learned ones in the past, it would be difficult for the subjects to harness the robot to keep it in the right directions. In the usual RNN learning, if the contents of the current memory conflict with the one to be newly learned, the internal structure of the RNN could be deflected severely and undesired pseudo memories could be generated. Consolidation learning is beneficial in this aspect since this scheme allows iterative rehearsing of past experiences. Also, training with enough number of rehearsed sequences could achieve a sort of generalization while attaining the global structures in the internal representation.

6 Conclusion

In this paper, we showed that incremental learning is essential and important for man-machine cooperation. We also pointed out that it is difficult to actualize context dependence learning. The RNN was introduced as a learning algorithm system which can treat the hidden state problem. The target task was the human navigation by a humanoid robot called Robovie. The FFNN and the RNN were compared as the learning algorithm of Robovie. Although the performances of the RNN and the FFNN both showed improvements in the early stages of learning, they gradually became unstable as the subject changed the operation by learning on its own. Finally, all performances failed. The results obtained when the consolidation-learning algorithm, which uses the rehearsal outputs of RNN, was applied confirmed that this algorithm's performance was improved even when interactive learning continues for a long time and that the subject's mental impressions were better. The analyzing and comparing the characteristics of RNNs produced results which support these differences.

Two further studies should be carried out. One should be a more detailed analysis of the characteristics of the consolidation-learning algorithm. Although the robustness of the RNN has been compared in this paper, the relation between the dynamic structure of the RNN and the learning algorithm has not been analyzed mathematically yet. The other study that should be carried out is a transition structure analysis of the cooperation form between a human and a machine in the interactive learning process. Although we showed that the RNN with the consolidation-learning algorithm could continue to improve the performance, the adaptation phenomenon that might be the changing process of the cooperation form has not been analyzed in detail yet. The correspondence between the transition of the RNN structure and the development of human operation should be investigated.

References

1. Y. Hayakawa, I. Kitagishi, Y. Kira, K. Satake, T. Ogata, and S. Sugano: An Assembling Support System based on a Human Model -Provision of Physical Support According to Implicit Desire for Support, Journal of Robotics and Mechatronics, Vol.12, No.2 (2000) 118-125
2. T. Sawaragi, T. Kudoh, and S. Ozawa: Extracting Motion Skills from Expert's Proficient Operation Records Using Recurrent Neural Network, Reprints of 14th World Congress of IFAC, Beijing, Vol. M (1999) 359-364
3. Y. Miwa, S. Wesugi, C. Ishibiki, and S. Itai: Embodied interface for emergence and co-share of 'Ba', Usability Evaluation and Interface Design, in Proc. of HCI International 2001 (2001) 248-252
4. Y. Miyake, and T.Minagawa: Internal observation and co-generative interface, in Proc. of IEEE International Conference on Systems, Man, and Cybernetics, I229/I-237 (1999)
5. H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, R. Nakatsu: Robovie: an interactive humanoid robot, International Journal of Industrial Robotics, Vol. 28, No. 6 (2001) 498-503
6. L. Lin, and T. Mitchell: Efficient Learning and Planning within the Dyna Framework, in Proc. of the Second International Conference on Simulation of Adaptive Behavior (SAB'92) (1992) 281-290
7. J. Tani: Model-based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective, IEEE Trans. on System, Man and Cybernetics Part B (Special Issue on Robot Learning), Vol.26, No.3 (1996) 421-436
8. M. Jordan: Attractor dynamics and parallelism in a connectionist sequential machine, in Proc. of the Eight Annual Conference of the Cognitive Science Society (Erlbaum, Hillsdale, N.J) (1986) 513-546
9. D. Wolpert, and M. Kawato: Multiple paired forward and inverse models for motor control, Neural Networks Vol. 11, 1317-1329
10. J. Tani: An Interpretation of the 'Self' from the Dynamical Systems Perspective: A Constructivist Approach, Journal of Consciousness Studies, Vol.5 No.5-6 (1998)
11. D. Rumelhart, G. Hinton, and R. Williams: Learning internal representation by error propagation, In D.E. Rumelhart and J.L.Mclelland, editors, Parallel Distributed Processing (Cambridge, MA: MIT Press) (1986)
12. S.G. Hart et. al: Development of NASA-TLX: Results of empirical and theoretical research, InP.A.Hancock and N.Meshkati(eds.), Human Mental Workload, North-Holland (1988) 139-183

Stereo Camera Handoff

Kaimin Yuan[1] and Haiyi Zhang[2]

[1] Department of Computer Science, Dalhousie University
Halifax, Nova Scotia, Canada B3H 1W5
[2] Jodrey School of Computer Science, Acadia University
Wolfville, Nova Scotia, Canada B4P 2R6

Abstract. To track multiple moving objects in multiple surveillance cameras is a non-trivial problem, especially when occlusion is present. This problem is also referred as "Camera Handoff" as it hands over object's identification from one camerea to another camera, which shares a considerable common field of view with the first. In this paper, moving object handoff is accomplished using geometric constraints between the coordinate system of the two stereo cameras. We proposed a Two-Pass Matching Method to deal with the occlusion problem explicitly.

Keywords: Computer Vision, Stereo Vision, Video Surveillance, Camera Handoff, Multi-camera Tracking

1 Introduction

One of the underlying problems in multi-camera tracking is to associate objects in different camera views [3, 5]. One needs to establish correspondences between objects captured in each camera. This task is also referred as "Camera Handoff". "Handoff" approaches can be divided into geometric-based and recognition-based [6]. The former transforms the geometric features (normally the coordinates of object's location) to the same spatial reference frame before matching is performed. The latter is a special case of object recognition. Recognizable features are extracted from different cameras' views as models upon which to match objects. In the geometry-based domain, the presenting methods are categorized further into two classes, namely, "with terrain knowledge" and "without terrain knowledge." In the former situation, prior knowledge of the terrain information about geolocation and elevation is known. In the case that the terrain knowledge is unknown, geometric constraints must be applied to determine the relationship between cameras' views. Three kinds of constraints are currently present: "homography constraint [2, 5]," "epipolar constraint [4]," and "FOV (field of view) constraint [7]."

Generally, widely distributed cameras (typically used in outdoor surveillance applications) have totally different perspective views and often have different illumination environments due to their geographical dispersion. Thus, the recognition-based features such as colour, size and shape would be less reliable than the geometric features (typically the location in a common coordinate frame). However, geometric location matching inherently depends on the simultaneous visibility of an object to both cameras. This implies that only those objects within the overlapping field of

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 445-452, 2004.

view could be “handed off” correctly using geometry-based methods. The task in our research is to solve the problem: “when an object O moving from camera A into another adjacent camera B within their overlapping field of view, how can camera A handoff O’s identification to camera B, based only on geometric information?”

2 Proposed Solution: A Two-Pass Matching Method

Past methods, for examples, geometric method with terrain knowledge and geometric method using homography constraint etc [1, 2, 5], which can solve the problem mentioned in the section one. But all the above methods share the same limitation: they cannot handle the occlusion problem explicitly, which is a very common phenomenon in real world tracking applications. In the following, a new method is proposed to handle the handoff problem with the presence of occlusion under the configuration that two cameras are installed containing a considerable common field of view where the handoff is conducted. This solution works with the following assumptions:

- There exists a dominant ground plane in the monitored site and moving objects (usually moving persons) are in contact with this ground plane.
- There is no blind area in the common field of view. In other words, at least one camera will fully observe the moving object. This can be achieved by adjusting the camera placement.
- A single camera motion tracking module (which has been implemented in our previous work) correctly detects and tracks moving objects in both cameras locally. Each object holds a unique local identification and a bounding box tightly enclosing it.

2.1 The Homography’s Difficulty with Presence of Occlusion

The homography is a powerful constraint when there is a dominant planar ground in the scene. In that case, by looking at the “feet” of the object (middle location of the bottom of the bounding box), a simple homography transformation will help to locate the object’s position in the other camera’s FOV. Given a planar ground and correct “feet” position in both camera views, this kind of one-to-one correspondence provides great confidence. However, even with the assumption that the ground is planar, it is still difficult to guarantee the correctness of the “feet” position in the image. When the object’s “feet” are occluded by a static scene structure or even another moving object, the motion detection module reports a fake “feet” position, most likely the “belly” of the object (static occlusion) or another object’s “feet” (mutual occlusion).

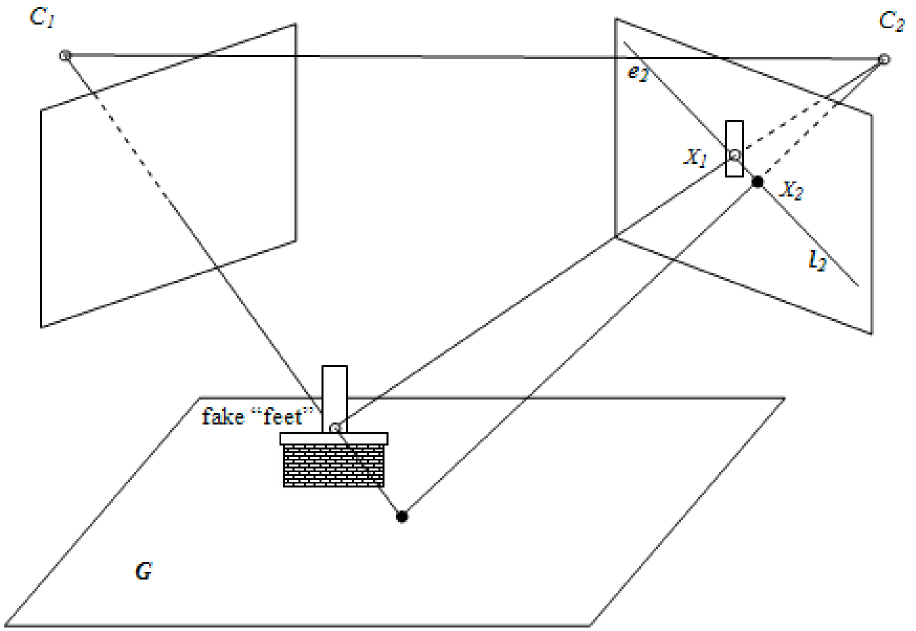


Fig. 1. (Scenario 1) When there is a static scene structure occluding a moving object in camera 1's view, an error may occur while performing the homography transformation.

In the first scenario (Fig. 1) where there is a static occlusion in camera 1's view, the object's location will be incorrectly transformed to X_2 instead of X_1, where the object is actually located in camera 2. In camera 1's view, what is observed is that the object's "feet" are located at the white point, which is actually its "belly." However, it is incorrectly assumed that this "belly" position is on the ground plane, so the homography transformation will indicate that the object's "feet" position should be at X_2 in camera 2's view, where probably no object is actually located. Note that X_1 and X_2 must be lying on the same epipolar line, because the observed "feet" in camera 1 and the assumed "feet" position is actually collinear with the optical center C_1 in the three dimensional world coordinate system.

In another scenario, as shown in Fig. 2, two objects occlude each other in camera 1, but camera 2 observes them separately. (As for the situation that two objects occlude each other in both camera views, there is no pure geometry-based method to solve this problem. So no match will be made until the two objects separate, at least in one camera's view.) In this case, camera 1 will report one merged object and camera 2 will report two separate objects. The tracking data of camera 1 will tell which objects this merged one may possibly contain, for example, O_1 and O_2. Heuristically, it is assumed that the merged object share the same "feet" position with the closer object (to the camera in the 3D world, as O_1 in Fig. 2), because the closer object (O_1) must be lower than the further object (O_2) in camera 1's image plane, given that O_1 and O_2 have comparable scale of size and both of them are far from the camera, which is true in a typical surveillance configuration. Since O_1 occludes O_2's feet, the homography transformation will only establish correspondence between the merged object to what O_1 is observed as in camera 2, but leaving O_2 in camera 2 unsolved.

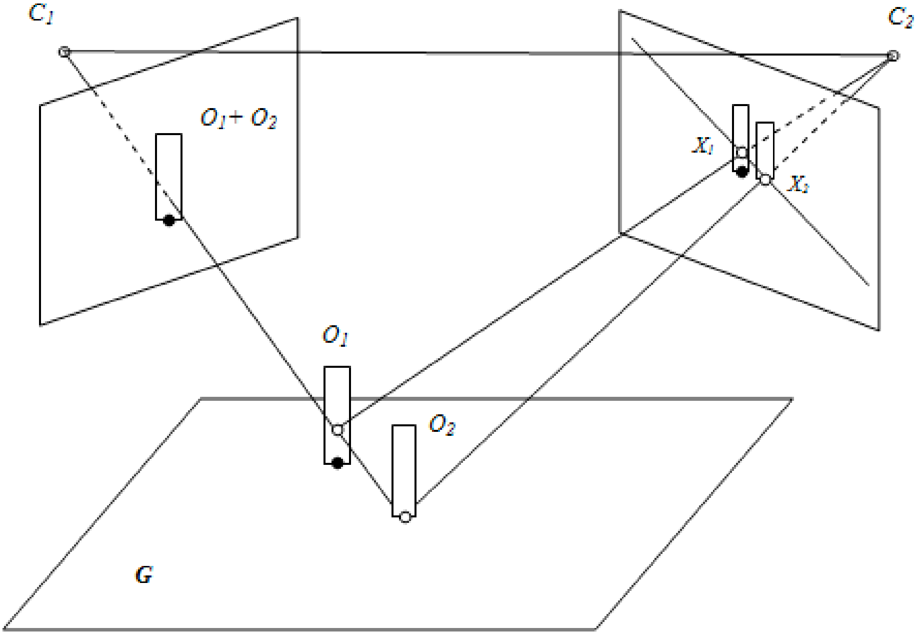


Fig. 2. (Scenario 2) Simple homography transformation cannot handle the complexity of mutual occlusion between moving objects. (O1 occludes O2 in camera 1)

According to the above analysis, the incorrectness of "feet" location with the presence of occlusion leads to the failure of simple homography coordinate transformation, so a new method is proposed to address this problem.

2.2 A Two-Pass Matching Method

The new method still applies homography as its primary tool to conduct matching, however, the matching strategy is carefully designed to target the occlusion problem.

The method has the following steps:

1. Initialization:

   After mounting the cameras, estimate the geometric parameters: the homography matrix associated with the ground plane and the fundamental matrix associated with the two cameras. The homography matrix and fundamental matrix can be calculated by finding enough correspondence points in both views [4, 5]. This can be either accomplished by landmark points or automatic point correspondence establishing algorithms.

2. First Pass Matching:
   Start online tracking.

   If an independent object $U_i^1$of camera 1 (not occluded by another object) enters the overlapping area, get its feet position and transform it into the coordinates of the other camera's view according to the homography. If there is an object standing at that location in camera 2's view and there are no other ambiguous matches, it is said to be a handoff. After the first pass, all one-to-one matching objects will be solved with great confidence. In fact, these objects should be those independent objects perfectly viewed by both cameras without any occlusion.

   If two independent objects of the first camera match the same object in the second camera, it implies that camera 1 observes these two objects separately, while camera 2 sees them occluding mutually. In this case, handoff the identifications of these two objects to that merged object in camera 2. Keep tracking and solve the one to one correspondence when the two objects separate in camera 2's view.

   If a non-independent object M^1 (two objects occluding each other) in the first camera enters the common field of view, a match should be found in camera 2, as discussed earlier (Fig. 2). However, the correct assignment should be that M^1 has two matches in camera 2, assuming that the two objects are separate in camera 2's view. From tracking data in camera 1, it could be known which object is getting occluded and which one is occluding the other. Handoff the identification of that occluding object (on the top) to the matching object found in camera 2. The other unsolved object (the one being occluded) will be solved in the second pass.

3. Second Pass Matching:

   After the first pass matching, for each unsolved object U_i^1 of camera 1, three possibilities exist. The first possibility is that because of static occlusion, the "feet" position of U_i^1 is incorrectly detected in the first camera. In this case, search all unsolved objects U_j^2 of camera 2 across the corresponding epipolar line of this fake "feet" in camera 2, since all possible matches must be located across this line (See Fig. 1). For each candidate object U_j^2 along this line, project its "feet" coordinates back to camera 1 using the inverse matrix of the homography. If this U_j^2 is the true match of U_i^1, the back-projected "feet" position in camera 1 should be the true "feet" of U_i^1 but getting occluded by some static structure. According to this heuristic, a matching pair (U_i^1, U_j^2) can be solved by checking if this back-projected "feet" of U_j^2holds a close horizontal coordinate with the fake "feet" of U_i^1but is a little bit lower vertically (within a certain threshold).

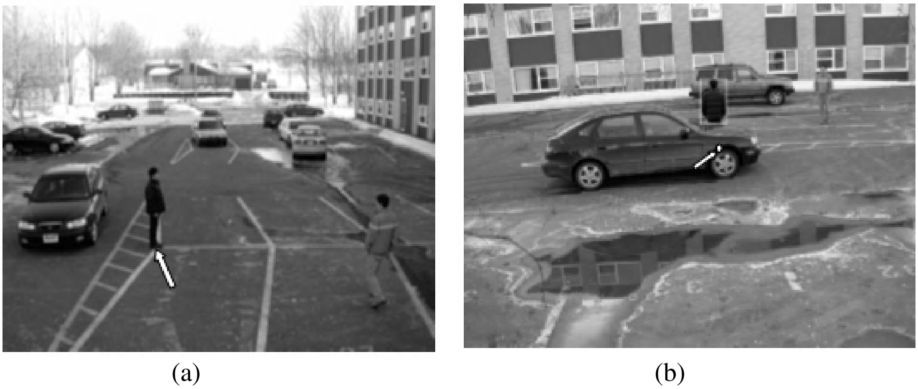
   The second possibility is that U_i^1 is fully observed in camera 1 but gets occluded by static features in camera 2. Similarly, project the "feet" position of U_i^1 using homography to camera 2's view. The projected "feet" should be a little bit lower than the fake "feet" of U_j^2 in camera 2, if (U_i^1, U_j^2) is a true match.

   The third possibility is that the unsolved object of camera 1 is that occluded object B^1 (the bottom one) in a non-independent object M^1. The occluding object T^1 of M^1 (the top one) has been solved in the first pass, as described before. In this case, search all the unsolved objects U_j^2 of camera 2 and back-project the "feet" of U_j^2 to camera 1's view. If this back-projection is located within the bounding box of M^1, it can be stated that (B^1, U_j^2) is a match.

4. On a per-frame basis, repeat the above two-pass matching strategy until the object leaves the first camera's view. For each frame, get a vote on a specific object identification assignment. After counting all the votes over time, the most likely handoff assignment is obtained.
5. For all remaining unsolved objects, the system will propose them to a human monitor for a final solution.

3 Experimental Results

In order to prove the feasibility of the proposed two-pass algorithm, two experiments were conducted regarding the two different scenarios, namely, the static occlusion and the mutual occlusion. Homography matrix and fundamental matrix are estimated by finding enough hand-crafted corresponding points in both camera views. Experiments are performed in an off-line configuration without being fully applied to a full-rate video sequence. In another word, only static snapshots (for both cameras) of the "handoff" instants were taken. Fig. 3 illustrates the results of the first experiment. A moving object is occluded by some static structure (the car) in one camera's view (b) but is fully visible to another camera (a). By the proposed method in Section 2, the homography transformation is performed from the "feet" position in (a) to that in (b), which are pointed by the arrows in (a) and (b), respectively. Above (within a certain threshold) from the inferred "feet" position in (b), an object is found, which can then be explained as the object that is getting occluded.

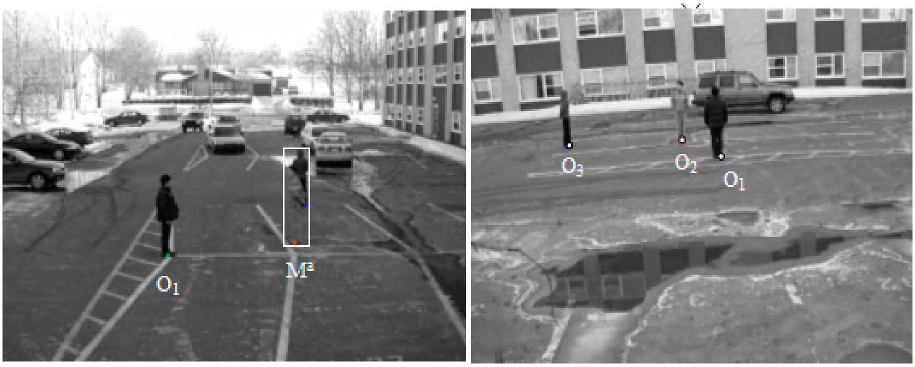


(a) (b)

Fig. 3. (Experiment 1) Object is occluded by static structures in one camera but fully visible in another camera.

The experiment 2 tackles the second scenario, namely, the mutual occlusion. As illustrated in Fig. 4 (a), two moving objects are mutually occluded but in (b) they are fully visible. The match between O_1^a and O_1^b can be solved either by the homography projection from (a) to (b) or the back-projection from (b) to (a), with no ambiguity.

Notice that M^a is a merged object, which is known from the tracking history of camera (a). By a homography transformation, the "feet" of M^a can be matched to the "feet" of O_2^b. The object O_2^b can then be explained as the occluding object (which is closer to the camera) in M^a of camera (a). Since the back-projected position of the "feet" of O_3^b from (b) to (a) is within the bounding box of (the "belly" of) M^a, the object O_3^b can be matched to the object which is getting occluded in the merged object M^a.



(a) (b)

Fig. 4. (Experiment 2) One moving object is occluded by another moving object in one camera but fully visible in another camera.

4 Remarks

In this paper, the sub-problem of stereo camera handoff is tackled by exploiting geometric constraints to estimate the relationship between the coordinate systems of two adjacent stereo cameras, within the overlapping field-of-view. First, a survey on this particular problem was conducted. Inspired by the existing state-of-the-art, a two-pass matching algorithm is proposed as the solution, which is validated by positive experimental results. The experiments were conducted in an off-line mode; hence it didn't provide confidence measure in an on-line environment. Future work will integrate existed single-camera tracking module with the proposed two-pass matching method, which fuses the tracking data of both single cameras and presents global decisions; and complete experiments will then be performed in an on-line configuration to fully evaluate this proposed method.

References

1. O.D. Faugeras: Three-Dimensional Computer Vision. MIT Press (1993)
2. Greg Kogut, and Mohan Trivedi: A Wide Area Tracking System for Vision Sensor Networks. 9th World Congress on Intelligent Transport Systems, Chicago, Illinois (2002)

3. Ismail Haritaoglu, David Harwood, and Larry S. Davis: W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. 3rd International Conference on Face and Gesture Recognition, Nara, Japan (1998)
4. Richard I. Hartley: In Defense of the Eight-Point Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence (1997) vol. 19 580-593
5. Lily Lee, Raquel Romano, and Gideon Stein: Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame. IEEE Transactions on Pattern Analysis and Machine Intelligence (2000) vol. 22. 758-767
6. Omar Javed, Sohaib Khan, Zeeshan Rasheed, and Mubarak Shah: Camera Handoff: Tracking in Multiple Uncalibrated Stationary Cameras. Workshop on Human Motion (HUMO'00) (2000) 113
7. S. Khan, O. Javed, and M. Shah: Tracking in Uncalibrated Cameras with Overlapping Field of View. 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (2001)

Word Separation in Handwritten Legal Amounts on Bank Cheques Based on Spatial Gap Distances

In Cheol Kim, Kyoung Min Kim, and Ching Y. Suen

Centre for Pattern Recognition and Machine Intelligence (CENPARMI),
Concordia University, 1550 de Maisonneuve Blvd. West, Suite GM606,
Montreal, H3G 1M8, Canada
{kiminc, kkm, suen}@cenparmi.concordia.ca

Abstract. This paper presents an efficient method of separating words in handwritten legal amounts on bank cheques based on the spatial gaps between connected components. Currently all typical existing gap measures suffer from poor performance due to the inherent problem of underestimation and overestimation. In order to decrease such burden, a modified version for each of those existing measures is explored. Also, a new method of combining three different types of distance measures based on 4-class clustering is proposed to reduce the errors generated by each measure. In experiments on real bank cheque database, the modified distance measures show about 3% of better separation rate than their original counterparts. In addition, by applying the combining method, further improvement in word separation was achieved.

1 Introduction

Automatic recognition of legal amounts on bank cheques has been intensively studied for many years in order to minimize manual efforts in bank cheque processing [1, 2]. However, due to the shape variability of the writers' unconstrained writing style, and horizontal overlapping, touching, and irregular gaps between connected components limited by space on bank cheques, legal amount recognition still remains as a challenging task. Separating a sentence into words and analyzing them is a typical approach to recognize a legal amount. In this case, precise extraction of the individual words is an essential step for achieving a reliable recognition performance.

In many related studies, measuring and sorting the spatial gaps between connected components using a specific distance measure has been employed as a typical approach to extract words. Seni and Cohen [3] proposed eight distance measures for extracting words from a handwritten text line. Among them, bounding box (BB) method that computes the horizontal distance between the bounding boxes of two adjacent connected components, and run-length/Euclidean with heuristics (RLEH) method that estimates the gap between two connected components using either the minimum run-length or the minimum Euclidean distance have shown superior performance. Mahadevan and Nagabushnam [4] proposed a convex hull (CH) method that approximates the gap between components by the distance between the convex hulls surrounding each component.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 453-462, 2004.

The above three distance measures, BB, RLEH, and CH methods are simple and quite efficient but all suffer from the inherent problem of underestimation or overestimation. In order to reduce their drawbacks we first propose to modify each of these measures. In the case of BB method, the left and right boundaries of bounding boxes are adjusted appropriately to reduce the underestimation errors. Such concept of adjusting boundaries is also applied to the RLEH method to blunt its over-sensitivity to component shape caused by dealing with component contours directly. In CH method, a pair of convex hulls each of which respectively encloses the left and right part of a given connected component is newly introduced to avoid an overestimation problem. Next, a new method of integrating these three distance measures based on 4-class clustering technique is proposed to compensate effectively the errors in each measure, whereby further improvement in performance of word separation is expected.

In word separation experiments on a legal amount database extracted from real-life bank cheques, we demonstrate the effectiveness of the modified distance measures and combining method based on 4-class clustering by comparing their separation rates with those of the original distance measures.

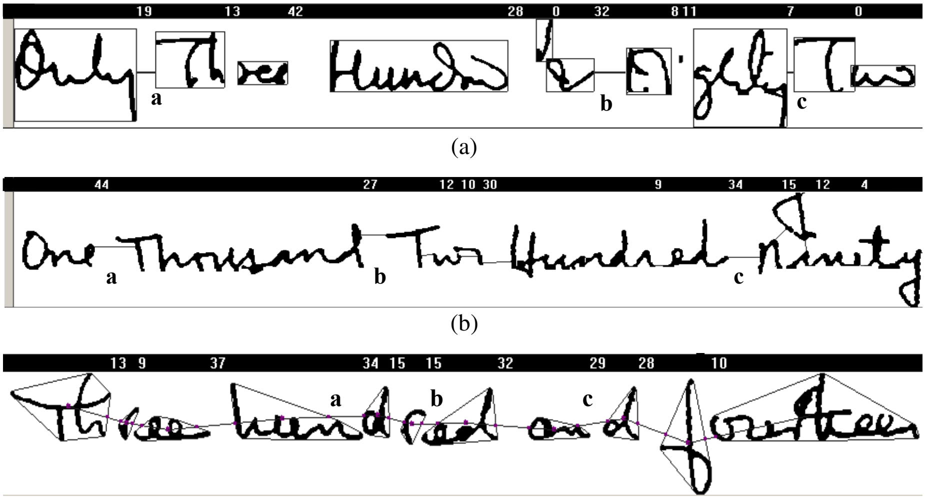
2 Distance Measures for Gap Estimation

For a word separation task, we first employ three well-known distance measures, BB, RLEH, and CH method, and investigate their geometrical properties and drawbacks. Next, a modified version of each measure for reducing their estimation errors is introduced.

2.1 Distance Measures

For gap estimation, the BB method simply computes the length of a horizontal line between the smallest rectangles respectively enclosing two adjacent connected components, as shown in Fig. 1 (a). If the bounding boxes overlap horizontally, i.e. the right boundary of the left box extends to the left boundary of the right box, the distance is considered as zero. The RLEH method, shown in Fig. 1 (b), uses the minimum run-length or the minimum Euclidean distance with some heuristics to estimate the gap between a given pair of connected components. If the connected components overlap vertically by more than a threshold determined heuristically, the minimum run-length is used, and the minimum Euclidean distance, otherwise. In the CH method, the smallest convex polygon, called convex hull, enclosing each connected component should be estimated first. Next, we obtain the particular points where two adjacent convex hulls are intersected by the line linking their centroids. Finally, the gap between two connected components is defined as the Euclidean distance between these intersection points as shown in Fig. 1 (c).

These distance measures provide a simple gap estimation but frequently they produce serious estimation errors like underestimation or overestimation. The BB method suffers from underestimation due to the rectangular nature suppressing component shape to a box. As can be seen from Fig. 1 (a), the gap between the connected components, at least one of which has horizontally protruded ‘head’ or ‘tail’ part (see gaps **a**, **b**, **c**), is usually underestimated. The RLEH method estimates gaps directly from component contours. In such case, estimation results are very sensitive to component shape and vertical positioning of components, as shown in Fig. 1 (b) (see gaps **a**, **b**, **c**). The CH method also suffers from an overestimation problem when an ascender or a descender is included in and located on either the beginning or the end part of connected component, as shown in Fig. 1 (c) (see gaps **a**, **b**, **c**).



(a)

(b)

(c)

Fig. 1. Gap estimation using (a) BB, (b) RLEH, and (c) CH method

2.2 Modified Distance Measures

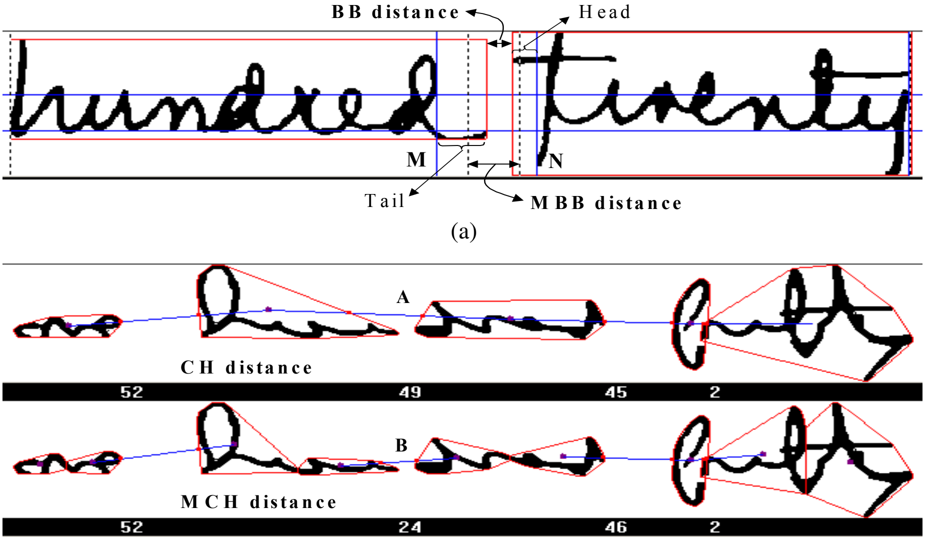
Next, we modify the above distance measures to avoid their overestimation or underestimation errors. In the case of BB method, we adjust the left and right boundaries of the bounding box of the component containing horizontally protruded ‘head’ or ‘tail’ part. Considering that such protruded ‘head’ and ‘tail’ parts usually consist of a single horizontal stroke as shown Fig. 2 (a), the ‘head’ part is defined as the region from the leftmost position to a node point (denoted by ‘N’) where the value of horizontal histogram is larger than βW. Here, W is the average stroke thickness in the entire image calculated using a simple mathematical method proposed in [5], and β is empirically determined as 1.25, considering some possible noise or distortion on a stroke. Similarly, the ‘tail’ part is defined as the region from the rightmost position to the node point (denoted by ‘M’). Then, the new left and right sides of the bounding box are defined as follows:

$$L_x^{\text{new}} = L_x^{\text{old}} + \alpha H_{\text{wd}} \tag{1}$$

$$R_x^{\text{new}} = R_x^{\text{old}} - \alpha T_{\text{wd}} \tag{2}$$

Here, H_{wd} and T_{wd} denote the widths of the 'head' and 'tail' parts, respectively. And α is 0.35 if these parts are located within the body area and 0.25, otherwise. As can be seen from Fig. 2 (a), the gap between two connected components each of which has horizontally protruded 'head' or 'tail' part is quite well estimated by employing the modified bounding box (MBB).

The RLEH method is also modified (MRLEH) slightly based on such concept of adjusting bounding box to reduce its over-sensitivity to 'head' and 'tail' parts; the component contour located within adjusted bounding box is only considered for gap estimation. Unlike the MBB method, the parameter α is assigned as 0.25 or 0.2 according to the vertical positions of the 'head' and 'tail' parts. Furthermore, one more heuristic is introduced to deal with some special components like the dot in character 'i' or 'j', and broken part of a character; if two connected components do not overlap vertically, the gap between them is estimated by the bounding box distance instead of the run-length or Euclidean distance.



(a)

(b)

Fig. 2. Gap estimation using (a) modified BB distance and (b) modified CH distance

Lastly, in order to deal with the overestimation problem in the CH method, a pair of convex hulls for each connected component is introduced. We divide vertically a connected component into two equal parts and estimate two convex hulls enclosing each divided part. The gap between two adjacent components is then computed by considering two convex hulls enclosing the right part of the left component and the left part of the right component, as can be seen from Fig. 2 (b). This modified CH (MCH) method is quite attractive because it produces a reasonable gap distance even though an ascender or a descender is located on either the beginning or the end part of component whereby the gap is usually overestimated in the original CH method (compare gap **A** with **B**). Additionally, like the MRLEH method, a heuristic condition for dot or broken part is also added to the MCH method; if two connected components do not overlap vertically, the gap is defined as the horizontal distance between the intersection points of two convex hulls, instead of the Euclidean distance.

3 Experimental Results

In experiments, we investigated the effectiveness of the modified distance measures by applying them to the word separation task using 1030 image samples of legal amounts included in CENPARMI database called IRIS and comparing their performance to that of the original measures. The IRIS database was obtained from real-life bank cheques. Thus considerable noise, shape distortion, and space irregularity were present in the images. Before performing word separation, several preprocessing procedures including smoothing, slant detection and correction, and removal of non-character components such as line, comma, and numeral parts were conducted in advance.

3.1 Word Separation by 2-Class Clustering

To extract words from a legal amount image based on the spatial gap distance, we employ a clustering technique based on the LBG algorithm [6] that classifies all gaps within a given legal amount image into two classes: inter-character gap (ICG) and inter-word gap (IWG). Before clustering for word separation, some special image samples containing only ICGs or only IWGs are extracted first according to the following heuristic procedure.

(1) For a given legal amount image, calculate three types (based on BB, RLEH, and CH methods) of the maximum, minimum, and average gap distances.

(2) If the width from the leftmost connected component to the rightmost one is less than 15% of the entire width of image, all gaps are assigned as ICG.

(3) If that width is less than 35% of the image width, and at least two types of the maximum gap distances are less than the overall gap average obtained from the whole data set or all three types of the average distances are less than 70% of the overall gap average, all gaps are assigned as ICG.

(4) If that width is larger than 35% of the image width and at least two types of minimum gap distances are larger than the overall gap average, all gaps are assigned as IWG.

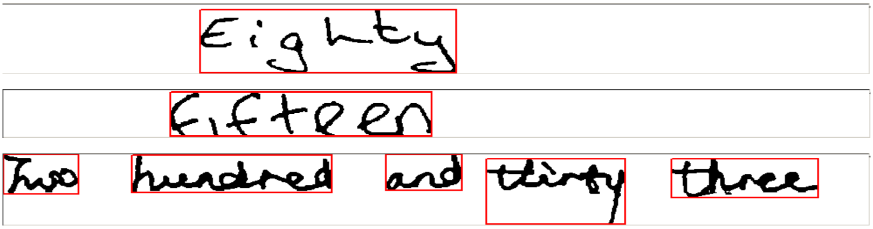
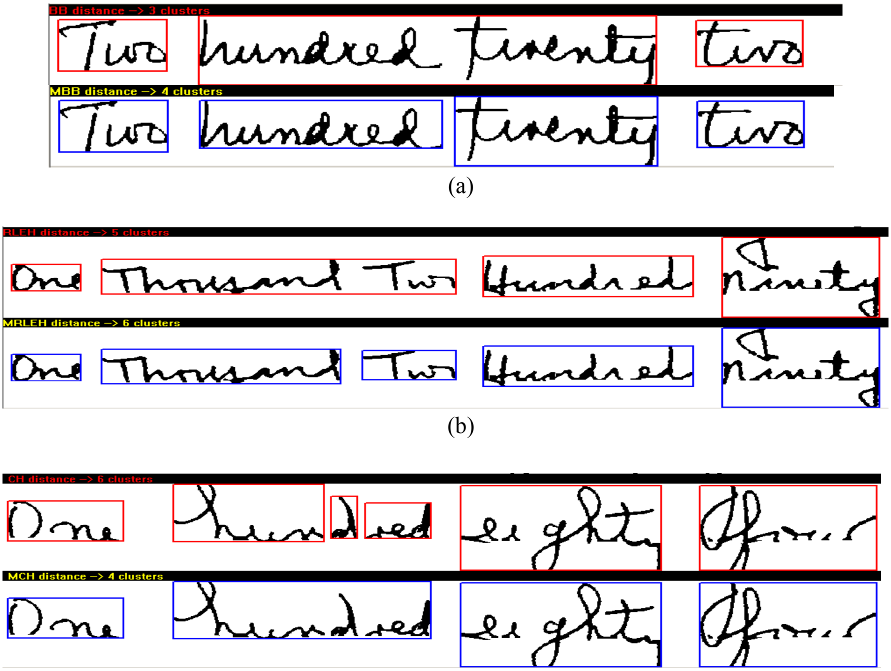


Fig. 3. Extracting words from image samples containing only inter-character gaps (ICG) or only inter-word gaps (IWG).

Figure 3 shows the examples of word separation for the image samples, in which only ICGs (1st and 2nd images) or only IWGs (3rd image) appear, using the above heuristic procedure. After extracting such special image samples, the clustering procedure based on LBG algorithm is then applied to the remaining samples. The experimental results in Table 1 show that the RLEH method performs best with its correct separation rate of 70.1% among the three original distance measures. Here, it should be noted that correct separation means that all words in a given legal amount image are perfectly isolated through the clustering procedure. Also, it can be found that all modified distance measures produced about 3% of better separation rate, when compared to their corresponding original distance measures.

Table 1. Experimental results of word separation

Distance Measures	BB	RLEH	CH	MBB	MRLEH	MCH
Correct Separation (%)	69.0	70.1	68.3	71.8	72.7	71.7


BB distance --> 3 clusters
MBB distance --> 4 clusters
RLEH distance --> 5 clusters
MRLEH distance --> 6 clusters
CH distance --> 6 clusters
MCH distance --> 4 clusters

(a)

(b)

(c)

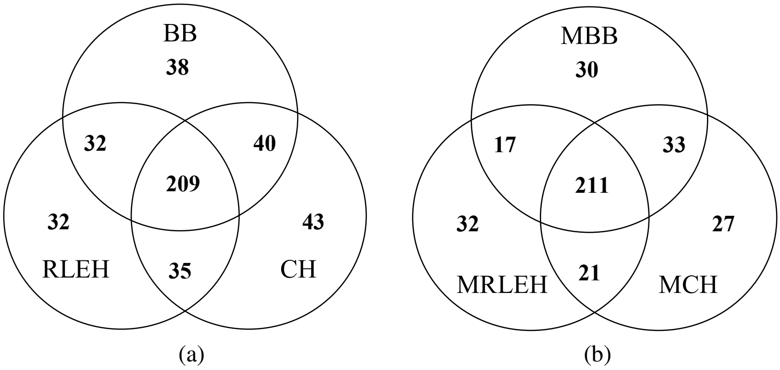
Fig. 4. Examples showing word separation error made by the original distance measures (upper part of each figure) and correction by their modified versions: (a) BB and MBB, (b) RLEH and MRLEH, and (c) CH and MCH

Figure 4 shows the examples of word separation illustrating the effectiveness of the modified distance measures. The words "hundred twenty" and "Thousand Two" shown in Fig. 4 (a) and (b) are misrecognized as one word (upper part of each figure) due to the underestimation by BB method and shape sensitivity by RLEH method, respectively. However, these parts are successfully divided into two words by using the modified measures, as shown in the lower part of each figure. In Fig. 4 (c), the word "hundred" is incorrectly divided into three parts due to the overestimation by the original CH distance but successfully merged by the modified method.

Next, we integrate these three different types of distance measures, thereby compensating the separation errors in each measure to achieve further improvement in word separation. A detailed description for our approach is provided in the following subsection.

3.2 Combining Three Distance Measures Based on 4-Class Clustering

The Venn diagrams shown in Fig. 5 represent the distribution of word separation errors due to the original BB, RLEH, and CH methods, and their modified versions, respectively. From these diagrams, it can be found that many errors are commonly produced from two or all of three distance measures even modified methods are applied. Thus a simple combining scheme such as the conventional majority voting is not expected to be effective in reducing such shared errors.



(a)

(b)

Fig. 5. Distribution of word separation errors in (a) BB, RLEH, CH methods and (b) their modified versions

In order to compensate effectively those errors commonly produced from two or all of distance measures as well as the errors caused by one measure only, we propose a new method of integrating three individual measures based on a 4-class clustering technique as follows:

(1) Estimate all gaps in a given image using a distance measure and divide them into 4 classes using LBG algorithm. The partitions are sorted in order of magnitude of their centroids.

(2) Assign an integer value, $\alpha \in \{2, 1, -1, -2\}$ to all the gaps according to the class to which they belong.

$$\alpha_i = \begin{cases} 2 & for \;\; g_i \in class \; 1 \\ 1 & for \;\; g_i \in class \; 2 \\ -1 & for \;\; g_i \in class \; 3 \\ -2 & for \;\; g_i \in class \; 4 \end{cases} \tag{3}$$

where, g_i denotes i-th gap in a given legal amount image. Thus, a larger positive value ($\alpha = 2$) is assigned to the gaps belonging to the class with a smaller centroid (*class* 1) to accentuate their possibility of ICG, and vice versa.

(3) For the i-th gap, three different integer values, α_i^{BB}, α_i^{RLE}, and α_i^{CH} are assigned according to the distance measures: MBB, MRLEH, and MCH.

(4) Define i-th gap as ICG if $\alpha_i^{BB} + \alpha_i^{RLE} + \alpha_i^{CH}$ is positive, and IWG, otherwise.

In the second part of word separation experiments on the proposed combining method, we achieved more than 75% of correct separation rate. This performance is much higher than those of any types of individual distance measures so far used in our experiment as well as those of other studies that used a similar database for evaluating their approaches, as shown in Fig. 6.

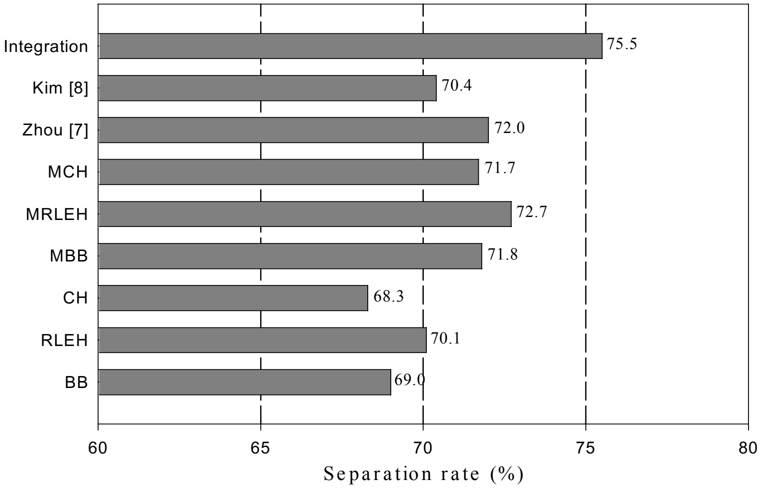


Fig. 6. Comparison of word separation rate according to methodologies

The Venn diagram in Fig. 7 clearly shows that the errors generated from only one of three distance measures are almost perfectly removed. Moreover, it can be found that the errors common to two of three distance measures are also reduced effectively. However, the number of errors common to all of three distance measures is hardly reduced even when the combining method based on the 4-class clustering is applied.

A primary assumption to use the spatial gaps between connected components for the problem of dividing a sentence into words is that the gaps between words are usually larger than those between characters. However, such assumption does not match well with the problem of extracting words from a legal amount on a bank cheque; an inter-word gap is not always larger than an inter-character gap, and horizontal overlapping and touching of two adjacent words arise frequently due to the writer's unconstrained writing style and writing space constraints. Accordingly, our analyses indicate that employing a prior knowledge such as possible maximum and minimum lengths of a word in lexicon or the number of possible words in a legal amount is needed to remove such troublesome errors. Moreover, introducing other methodologies such as implicit segmentation scheme or recognition based segmentation approach will be helpful to achieve further improvement in word separation performance.

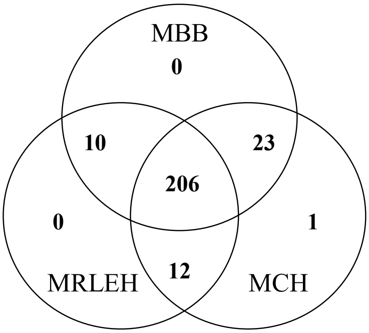


Fig. 7. Error distribution by combining method using 4-class clustering

4 Conclusions

Extracting words from the legal amount on a bank cheque has been performed based on the spatial gaps between connected components. The existing distance measures: BB, RLEH, and CH, are simple and quite efficient for gap estimation but all suffer from the inherent problem of underestimation or overestimation. To alleviate such problem, we have modified each distance measure; adjusted the left and right boundaries of the bounding box for the BB and RLEH methods, and introduced a pair of convex hulls enclosing equally divided connected components for the CH method. Furthermore, we proposed a salient method of integrating three individual measures based on 4-class clustering technique in order to effectively reduce the errors common to two of three distance measures as well as the errors made by each individual distance measure only.

Through a series of word separation experiments on CENPARMI IRIS database, we found that the modified measures show a better performance in terms of their separation rates compared with their corresponding original distance measures. Also, further improvement in performance of word separation was achieved by applying the combining method.

As a future study, we plan to introduce a priori knowledge about the lexicon of legal amount and to employ other methodologies such as implicit segmentation scheme or recognition based segmentation approach to reduce word separation errors caused by gap irregularity and overlapping or touching between words.

References

1. Guillevic, D., Suen, C.Y.: Recognition of Legal Amounts on Bank Cheques. Pattern Analysis and Applications, 1(1) (1998) 28-41
2. Kaufmann, G., Bunke, H.: Automated Reading of Cheque Amounts. Pattern Analysis and Applications, 3(2) (2000) 132-141
3. Seni, G., Cohen, E.: External Word Segmentation of Off-line Handwritten Text Lines. Pattern Recognition, 27(1) (1994) 41-52
4. Mahadevan, U., Nagabushnam, R.C.: Gap Metrics for Word Separation in Handwritten Lines. Proc. Int'l Conf. Document Analysis and Recognition, 1 (1995) 124-127
5. Schürmann, J.: Document Analysis – from Pixels to Contents. Proc. IEEE, 80(7) (1992) 1101-1119
6. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantizer Design. IEEE Trans. Communications, COM-28(1) (1980) 84-95
7. Zhou, J., Suen, C.Y., Liu, K.: A Feedback-based Approach for Segmenting Handwritten Legal Amounts on Bank Cheques. Proc. Int'l Conf. Document Analysis and Recognition, (2001) 887-891
8. Kim, K.K., Kim, J.H., Chung, Y.K., Suen, C.Y.: Legal Amount Recognition Based on the Segmentation Hypotheses for Bank Check Processing. Proc. Int'l Conf. Document Analysis and Recognition, (2001) 964-967

Shape Recognition of the Embryo Cell Using Deformable Template for Micromanipulation

Min-Soo Jang[1], Seok-Joo Lee[1], Ho-dong Lee[1], Yong-Guk Kim[2*]
Byungkyu Kim[3], and Gwi-Tae Park[1]

[1]Dept. of Electrical Engineering, Korea University, Seoul, Korea
[2]School of Computer Engineering, Sejong University, Seoul, Korea
ykim@sejong.ac.kr
[3]Korea Institute of Science and Technology, Seoul, Korea

Abstract. In biology, manipulating a micro-scale object such as chromosome, nucleus or embryo has been an important issue. For instance, skillful manipulation of the embryo cell in the biological experiment requires many years experience with a complex setup. Moreover, such process is usually very slow and requires many hours of intense operations such as trying to find the position of the cell within a petri dish and injecting a pipette to the cell from the best orientation. We have designed a new vision system, by which it finds the region of the mouse embryo cell, and then tracks the nucleus and the polar body within the cell, respectively, using the deformable template algorithm. Performance of the system is compared to the manual case.

1 Introduction

Manipulating an object within the micro scale is a key technology in biology, since the sizes of DNA, chromosome, nucleus, cell and embryo are within the order of micrometer [2], [3], [6], [10]. However, such manipulation is typically difficult, because the objects are normally weak, small (e.g. the size of the mouse embryo cell is about 1μm – 40μm), and the operation should be carried out in the slippery culture fluid. And yet, most of such micromanipulations are carried out manually. Therefore, the manipulators should often spend over a year to accomplish a biology experiment. Since they depend on visual inspection, the success rate and efficiency of manipulation is extremely low because of the eyestrain. For such reasons, automation of the micromanipulation has been asked.

Several approaches have been proposed to analyze bio cells using computer. One is to use template matching or morphological description of the cells (i.e. red blood cells and leukocyte) [1], [8]. In the other cases, utilization of visual features such as curvature, skeleton, and fractal dimension of the nucleus for leukemia diagnosis [9], or segmentation of color image for analyzing bone marrow image [12], and representation of peripheral blood smear [5] have been used, respectively.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 463-472, 2004.

In the embryo cell manipulation, the insertion position of the pipette is determined by the structure of the cell. In this paper, we propose a new scheme where a machine finds the embryo cell and then recognizes inner structures of the cell, such as nucleus, polar body using the deformable templates. Because the cell is alive and laying in the culture fluid, the conventional vision algorithms such as template matching and mathematical morphology operation show some limitation in recognizing the structure of the cells [1], [8]. Moreover, several factors such as diverse size and shape of living cells and optical characteristic of the culture fluid make it difficult to recognize the cell accurately. It is found that the deformable template method is the better choice [11]. We define first two deformable templates for the nucleus and polar body for a cell, respectively. And then, the calculated value of the energy function for each template will determine whether it is best fit or not. Therefore, the deformable template and the energy function allow us to recognize accurate position of inner structure of the cell.

For the micromanipulation, an optical microscope is typically used. In our setup, we utilize three images, acquired from three different magnification ratios of the microscope that maintain the same viewpoint. The lower magnification image is used for searching the ROI (Region Of Interest), whereas the higher magnification image for recognizing the inner structure of the cell. Utilization of multiple images increases the efficiency and the precision.

In section 2, the details of our micromanipulation system are described. The proposed algorithm for recognizing the embryo cell is discussed in section 3 and section 4. Result of experiments is described in section 5. Finally, we summarize our results, and discuss the performance of the whole system in section 6.

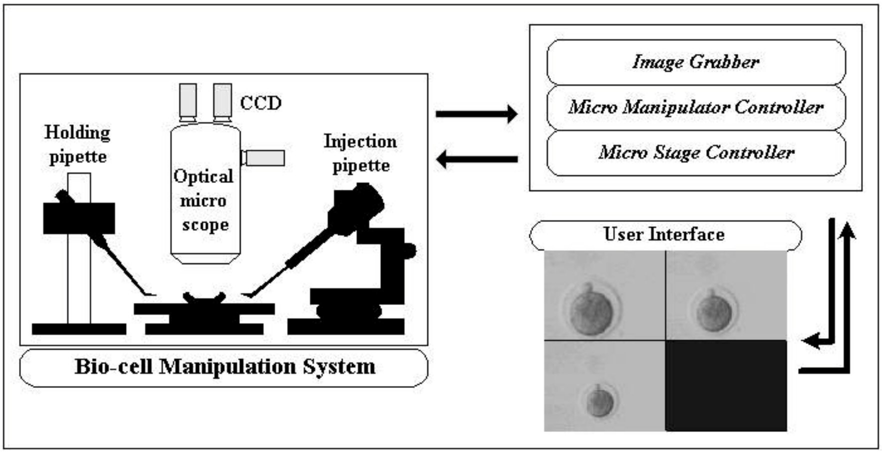


Fig. 1. A schematic view of the micromanipulation system

2 The Micromanipulation System

Our vision-based micro-manipulation system consists of three main parts: first, an optical stereomicroscope with three CCD cameras mounted on top of the microscope, secondly, the micro XY stage and micromanipulator for a holding pipette and an injection pipette, and thirdly, a PC with a controller for the micro XY stage, the micromanipulator and the image grabber. Fig. 1 shows the vision based micromanipulation system and its schematic architecture.

The system contains an optical stereomicroscope with three magnification ratios (x400, x600 and x1200). So, three images can be acquired simultaneously. The micro XY stage is driven by lead-screws for translation movement with a travel range of 25mm. Precision roller bearings guarantee straightness of travel within 2μm. It uses a compact closed-loop DC motor with a shaft-mounted high-resolution position encoder, and the precision gear provides 0.1μm minimum incremental motion with a resolution of 0.0085μm. Both the injection pipette and holding pipette are installed on the micromanipulator with 3DOF (Degree Of Freedom), actuated by three stepping motors. The mobile range of each axis is 25.4mm and the step resolution of the stepping motor is 0.04μm.

3 The Operation of the System

In this section, we describe how the system is operated, and the merit of using the multiple magnification images. By nature, as the magnified image provides more precise view, one can get better information of the target. However, the highly magnified image brings also narrower visual field. Moreover, any slight movement of holding or injection pipette conveys a large displacement in the image. On the other hand, since the less magnified image shows a wide area, one can find the cell within the image rapidly, and ignore small amount of displacement error. However, the lower resolution of such image makes it difficult to recognize exact position of inner structures of the cell. Fig. 2(a) shows a shot for the user interface. Three windows show images that have magnification ratio of 400, 600 and 1200, respectively. As you can see, the highly magnified one provides a more detail information about inner structure of the cell than the less magnified one.

The use of those multiple images increases efficiency and precision of the micromanipulation system. First, the system extracts the ROI within a given image with a magnification ratio of 400 by using histogram segmentation algorithm as described in section 4. When the system has succeeded in extracting the ROI area of the cell, it generates a moving trajectory to the center of the image by moving the micro XY stage using visual feedback. Then, it starts to search the inner structure of the cell. However, when it fails, it regards that cell as an irregular cell. Fig. 2(b) shows the flowchart of the micromanipulation system.

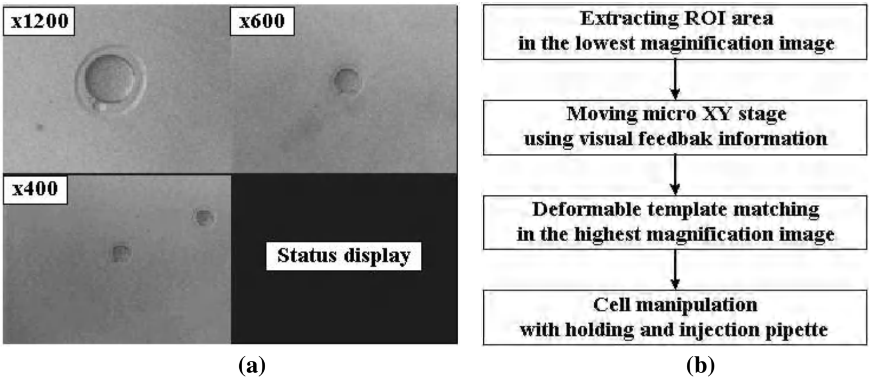

x1200
x600
x400
Status display
Extracting ROI area
in the lowest maginification image
Moving micro XY stage
using visual feedbak information
Deformable template matching
in the highest magnification image
Cell manipulation
with holding and injection pipette
(a)
(b)

Fig. 2. The user interface (a) and the flow chart (b) of the system

Our target here is the mouse embryo cell. It is widely used for the biology experiments, because of its similarity to the human embryo cell. Fig. 3 depicts the typical structure of a mouse embryo, consisting of nucleus, polar body and zona pellucida. Since we want to prevent any chance of destruction of the polar body during insertion of the injection pipette into the embryo, it is essential to recognize the exact positions of the nucleus and the polar body. Notice that the arrow A in Fig. 3 indicates the potential injecting direction of the pipette, which is perpendicular to the dotted line drawn over the nucleus and the polar body.

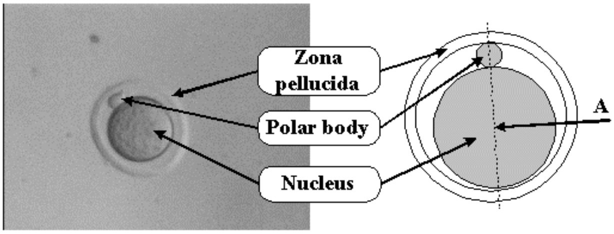


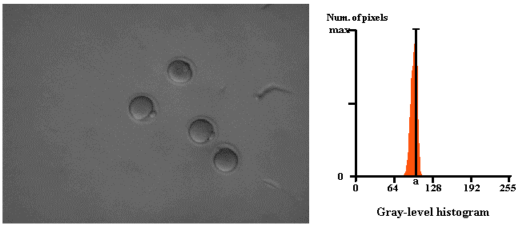
Fig. 3. Structure of embryo

4 Shape Recognition of the Mouse Embryo

In this section, we describe the embryo cell recognition algorithm. The present vision algorithm consists of two main parts: the ROI extraction and the cell recognition.

4.1 Extraction of the ROI

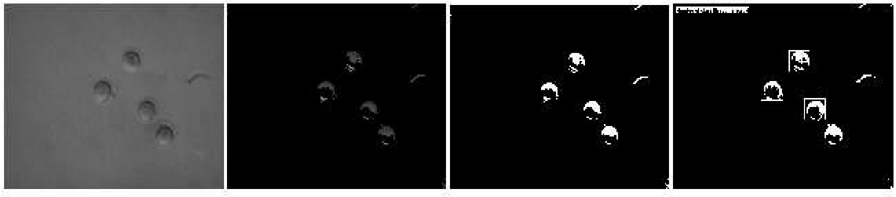
To recognize the shape of the cell, first of all, we need to extract the ROI within the acquired image. This process increases the recognition rate of the system and reduces the computational time by applying the deformable template method selectively to the ROI only. The process consists of the histogram segmentation and the nearest neighborhood clustering method [4]. Fig. 4(a) shows the mouse embryo cells embedded on the background, and Fig. 4(b) is the histogram for the same image. Since the background area occupies the majority of the image, the peak in the histogram corresponds to the background area. By using this property, we can easily eliminate the background area.



(a) **(b)**

Fig. 4. The histogram of the cell image

Fig. 5 explicitly illustrates several steps how the ROIs can be extracted within the given image. Once the background gray pixels are removed from the image, the image becomes Fig. 5(b). After digitization of that image, each cell area consists of a group of white pixels as shown in Fig. 5(c).



(a) **(b)** **(c)** **(d)**

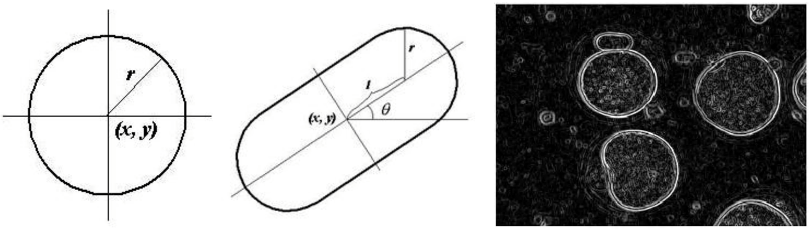
Fig. 5. Steps of determination of the ROI

The nearest neighbor algorithm finds the nearest pair of distinct clusters and merges the pair of clusters. The result image is shown in Fig. 5(d).

4.2 Deformable Template Model

The deformable template method models a target object using a template with a few parameters [7], [11]. It is used to recognize the target object by adjusting those parameters. Although one can have a complex deformable template using many parameters, then the search time will increase rapidly. Therefore the complexity of a deformable template model must trade off between the search time and the recognition accuracy. To simplify our task, we made three assumptions:

(a) Shape of the nucleus is similar to a circle.
(b) Shape of the polar body is similar to an ellipse.
(c) The nucleus and the polar body do not overlap each other.



(a) (b) (c)

Fig. 6. Two deformable templates and the potential field

4.2.1 Deformable Template for the Nucleus

Fig. 6(a) is the deformable template for the nucleus of the mouse embryo cell. It has three parameters, x, y and r, corresponding to two coordinates of the center and radius of the template, respectively. These parameters are selected for modeling the nucleus according to the assumptions (a). An arbitrary point on the template is defined as $p(x,y)$. From that point, the upper and lower contours are given by

$$\begin{aligned}&\textit{Upper contour } p(x,y)=\left\{x\in[-r,r]\mid y=\sqrt{r^2-x^2}\right\}\\&\textit{Lower contour } p(x,y)=\left\{x\in[-r,r]\mid y=-\sqrt{r^2-x^2}\right\}\end{aligned}\qquad(1)$$

4.2.2 Deformable Template for the Polar Body

Fig. 6(b) is the deformable template for the polar body of the mouse embryo cell. To model this template, we need the left and right half circles and two lines connecting two half circles. Hence, there are five parameters, x, y, l, r and θ, corresponding to two coordinates for the center, the length from the center of the template to the center of the half circle, radius of the half circle and the rotational degree of the template, respectively. We define an arbitrary point on the upper straight line which connects two half circle as follow

$$\begin{aligned}&p_{ul}(x,y):x\in[-l,l]\\&x=x\cos\theta+(y+r)\sin\theta\\&y=-x\sin\theta+(y+r)\cos\theta\end{aligned}\qquad(2)$$

where p_{ul} is an arbitrary point on the upper straight line. And an arbitrary point on the lower straight line, p_{ll} is defined as follow

$$\begin{aligned} &p_{ll}(x,y): x \in [-l,l] \\ &x = x\cos\theta + (y-r)\sin\theta \\ &y = -x\sin\theta + (y-r)\cos\theta \end{aligned} \tag{3}$$

Another arbitrary point on the left half circle, p_{lc} is defined as follow

$$\begin{aligned} &p_{lc}(x,y): x \in [-r,r] \\ &x' = -\sqrt{r^2 - y^2} \\ &x = (x'-l)\cos\theta + y\sin\theta \\ &y = -(x'-l)\sin\theta + y\cos\theta \end{aligned} \tag{4}$$

The final arbitrary point on the right half circle, p_{rc} is defined as follow

$$\begin{aligned} &p_{rc}(x,y): x \in [-r,r] \\ &x' = \sqrt{r^2 - y^2} \\ &x = (x'+l)\cos\theta + y\sin\theta \\ &y = -(x'+l)\sin\theta + y\cos\theta \end{aligned} \tag{5}$$

4.2.3 Energy Function

To determine whether the ROI is a nucleus or not, an energy function is introduced. The energy function is defined by equation (6). Here, as the template models the nucleus accurately, the total energy function becomes larger.

$$E_{total} = E_{edge} + E_{deviation} \tag{6}$$

The energy function consists of two terms: the edge enhancement term E_{edge} and the deviation term $E_{deviation}$. The first term is the summation of the potential fields for the coordinates on the template contour,

$$E_{edge} = \frac{1}{255n}\sum_{i=0}^{n}\phi_{edge}^{i}(x,y) \tag{7}$$

where n is the number of pixels on the template contour and the potential field ϕ_{edge} is the edge enhanced image by using *Sobel* operator [4]. The potential field ϕ_{edge} of an image of the mouse embryo cells is shown in Fig. 6(c), where one can observe a lot of noise, even after applying the noise elimination filter. Since such noise decreases the recognition rate, a deviation term is adopted. The deviation term $E_{deviation}$ is given by

$$E_{deviation} = 1 - \frac{1}{255}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(ave - \phi_{edge}^{i}(x,y))^2} \tag{8}$$

where ave is the average of the field potentials on the coordinates of the template contour. From the equation (8), notice that the deviation term is decreased when the edge potential is deviated from the given average value, whereas it approaches to the maximum value 1 as the average value is equal to the field potentials. By using the deviation term, we can increase the recognition rate and stability of the energy function.

4.3 Application of Deformable Templates to the Embryo Cell

Once an ROI is selected as shown in Fig. 5(d), the system starts to find the nucleus within the region, by scanning from the upper-left to the bottom-right, drawn as a gray rectangle in Fig. 7(a). The shape of the template is, of course, a circle since it is looking for a nucleus of the mouse embryo. The parameters for this template are two coordinates for the center and radius of the circle. The criterion for decision is based upon the total value of the energy function.

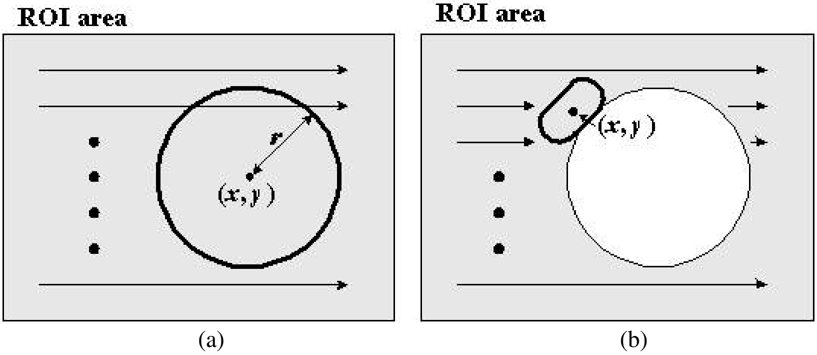


Fig. 7. Application of deformable template to the nucleus (a) and the polar body (b) within the ROI

As the nucleus of the cell is recognized, the next step is to find the polar body attached on the surface of the nucleus as illustrated in Fig. 7(b), since it is defined in the assumption (c) that the polar is not overlapped with the nucleus. In this case, the scanning is carried out similar to the above case except the nucleus area, which is already recognized.

5 Experiment and Performance of the System

We construct an embryo cell image database to evaluate the performance of the proposed embryo cell recognition algorithm. All images in our present database are acquired from the micromanipulation system described in section 2, and the database

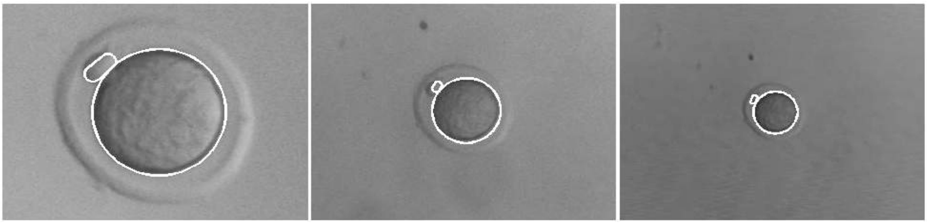


Fig. 8. Superposition of deformable templates to the embryo images

contains 138 mouse embryo images with 400, 600 and 1200 magnification ratio, respectively. Fig. 8 shows three image shots for an embryo cell, and two deformable templates (i.e. the circular one for the nucleus and the ellipse one for the polar body) are superimposed on each image.

Table 1 summarizes our result for each magnification ratio. It took 450ms, 100ms, 65ms at 1200:1, 600:1, and 400:1 ratio, respectively, suggesting that the larger magnification ratio, the longer the processing time. However, the recognition rate was roughly proportional to the magnification ratio, since it was 93.5%, 80.4%, 52.2%, respectively. Note that we counted it as a success case only if our algorithm recognized the nucleus and the polar body simultaneously.

Table 1. Processing speeds and recognition rates for different magnification ratios

Magnification ratio	Processing time	Recognition rate
1200:1	450ms	93.5%
600:1	100ms	80.4%
400:1	65ms	52.2%

Given that above data are measured for each image, we also look up how much time it required for accomplishing closed-loop case, such as finding cells, moving the microscope to the center of the image, and recognizing the structure of the embryo cell. These experiments were accomplished with living mouse embryo cells. Our result shows that the total time of such operations took about average 9 seconds. It was known that an expert operator can manipulate about 13~14 cells for a day, because he normally feels eyestrain and looses concentration. In contrast, the automatic micromanipulation system can manipulate many cells without any fatigue and losing any concentration.

6 Conclusions and Discussion

For developing an automatic biological micromanipulation for the mouse embryo cells, we have designed a new vision system where a deformable template algorithm was incorporated with the multiple view operation for recognizing the shape of the mouse embryo cell. Our strategy is to utilize multiple images that have same viewpoint but have multiple magnification ratio. The system uses the low magnification image to extract the ROI area within the image, whereas it utilizes the high magnification image to apply the deformable template model to the ROI to speed up the system.

The main part of the algorithm consists of the histogram segmentation, the nearest neighborhood clustering and a deformable template model. To increase the recognition rate and to reduce the computational load, a dynamic search area algorithm is designed. We have modified the deformable template method for the present task: one for the circular-shaped nucleus and the other for the elliptic-shaped polar body of the mouse embryo cell. For the stability and performance of the method, the energy function includes the deviation term as well as the edge potential field term. Result suggests that the present automatic system outperforms the manual case.

References

1. D. Anoraganingrum, "Cell Segmentation with median filter and mathematical morphology Operation", Proc of International Conference on Image Analysis and Processing, pp.1043-1046, 1999
2. F. Arai, A. Kawaji, P. Luangjarmekorn, T. Fukuda and K. Itogawa, "Three-dimensional bio-micromanipulation under the microscope", Proc. of IEEE ICRA, pp.604-609, 2001.
3. F. Arai, T. Sugiyama, P. Luangjarmekorn, A. Kawaji, T. Fukuda, K. Itoigawa, and A. Maeda, "3D bio-micromanipulation", International Symposium on Micromechatronics and Human Science, pp. 71-77, 1999
4. G. A. Baxes, "*Digital Image Processing: Principles and Applications*", John Wiley & Sons, Inc. 1994.
5. D. Comaniciu, P. Meer, and D.J.Foran, "Image-guided decision support system for pathology", Machine Vision and Application, vol. 11, pp. 213-224, 1999
6. Y. Kimura and R. Yanagimachi, "Intracytoplasmic Sperm Injection in the Mouse", Biology of Reproduction, Vol. 52, No.4, pp.709-720, 1995
7. G-S. Lee, "Automatic Face Region Detection Using Chromaticity Space and Deformable Template", Master thesis, Korea University, 2001.
8. X. Li, G. Zong and S. Bi, "Development of global vision system for biological automatic micromanipulation system", Proc. of IEEE ICRA, pp. 127-132, 2001.
9. D.M.U. Sabino, L.F. Costa, S.L.R. Martins, R.T. Calado, and M.A. Zago, "Automatic Leukemia Diagnosis", Acta Microscopia, vol. 12, Num. 1, 2003
10. S. Yu, and B. J. Nelson, "Microrobotics cell injection", Proc. of the IEEE ICRA, pp.620-625, 2001
11. A. Yuille, D. S. Cohen and P. W. Hallinan, "Feature extraction from faces using deformable templates", CVPR'89, pp. 104-109, 1989
12. X.W. Zhang, J.O. Song, M.R. Lyu, and S.J. Cai, "Extraction of karyocytes and their components from microscopic bone marrow images based on regional color features", Pattern Recognition, vol. 37, pp. 351-361, 2004

Improved Edge Enhanced Error Diffusion Based on First-Order Gradient Shaping Filter

Byong-Won Hwang[1], Tae-Ha Kang[2], and Tae-Seung Lee[1*]

[1] School of Electronics, Telecommunication and Computer Engineering, Hankuk Aviation University, 200-1, Hwajeon-dong, Deokyang-gu, Koyang-city, Kyonggi-do, 412-791, Korea
bwhwang@mail.hangkong.ac.kr, thestaff@hitel.net
[2] Agency for Defense Development, Yusoung P.O. Box 35, Yusoung-gu, Daejeon-city, Korea
thkang@add.re.kr

Abstract. The reconstruction of edge information using the prevailing error diffusion halftoning is represented as weak in visual and objective measuring criteria. In the previous work, an excellent improvement in reconstructing edge information of original image into belevel-toned image was achieved by adding an edge enhancing filter to the original error diffusion method. In spite of such effectiveness, the edge enhancing method made degradation in general information representing average tone of original image and even in edge information when excessive edge enhancing parameter is applied. In this paper, we seek a novel edge enhanced error diffusion method to preprocess original image so that further enhance edge information while preserving the good characteristic of the original error diffusion in general information. To confirm the achievement of the proposed method, we compare the method with both the standard error diffusion and the conventional edge enhanced error diffusion for Lena image in various objective measuring criteria.

Keywords: computer vision, digital halftoning, edge enhanced error diffusion, gradient preprocessing filter

1 Introduction

Since the 20th century, bilevel-toned image devices including fax machines, printers, and plasma display panels have developed rapidly and vastly. Although these devices usually have only the two levels of tones or colors in consideration of technical and economical points of view, the images output by the devices must be sawn as natural as possible. Digital halftoning has been introduced to content with such requirement.

Halftoning is the rendition of continuous-toned image into bilevel-toned one and let see the latter as the former when looked at from a distance. Of many halftoning algorithms studied before, the error diffusion proposed by Floyd et al. is remarkable for its superior blue-noise property [1]. It distributes the error made at a pixel over surrounding pixels by quantizing the pixel into bilevel tones and using an error diffusing filter that makes the average error for the entire image be zero.

[*] The authors contribute equally to the paper and are listed in alphabetical order.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 473-482, 2004.

However, the error diffusing filter is designed to retain the average tone of original image, i.e. the direct current element in frequency, so a degradation of original image for edge information of high frequency has to be made [2]. Bilevel-toned images face contradictory necessities. That is, they have to make the direct current elements of power spectrum for display error be zero in order to retain average tone of original image while having to minimize the error in power spectrum of high frequency in order to preserve original edge information.

The studies having been conducted to obtain a further achievement of the error diffusion include the methods to modify the error diffusion filter [3], to adaptively adjust filter coefficients to minimize local errors [4], to introduce the property of human visual system (HVS) [5], to utilize the characteristics of printers [6], and so on. Most of all, the edge enhanced error diffusion proposed by Eschbach et al. has singular performance. This method adds multiples of pixel tones to the original image in the process of error diffusion to emphasize edges of original images and get clearer bilevel-toned images. However, the bilevel-toned images converted by the method have some errors at the areas of low frequency because it uniformly applies the transformation to original images without considering local area characteristics. Along with it, as an excessively large multiple is added to emphasize edges of original image, it is even possible to distort the edge information.

The aim of this paper is to seek a novel error diffusion method to improve edge information while keeping up general information. The method measures gradients of pixels in original image and add the values of a function to process the measurements to intermediate modified image before a threshold to determine bileveled tones is applied in the standard error diffusion. The values of the function are large for edge areas and small for flat areas in original image, so the method provides both the good properties of the standard error diffusion and the edge enhanced error diffusion, and is able to further achieve enhancement in edge information. In this paper, the function is called the first-order gradient shaping of original image (FGSOI) filter.

The paper hereafter is organized as follows. In Section 2, we describe the edge enhanced error diffusion proposed in this paper. The performance of the proposed method is compared with those of the standard error diffusion and the existing edge enhanced error diffusion for various objective measuring criteria in Section 3. The comparison results are reported and discussed in Section 4 and we finally conclude the paper in Section 5.

2 Proposed Edge Enhanced Error Diffusion

The overall error diffusion system is depicted in Fig. 1. and the edge enhancing FGSOI filter is designated by the dotted box and the rest modules are the same as those of the standard error diffusion of Floyd et al. [1]. In the figure, $g(i,j)$ and $h(i,j)$ are the input image and the bilevel-toned image of $I \times J$ samples, respectively. It is assumed that $h(i,j)$ has 0 or 1 and $g(i,j)$ is belong to the range $[0,1]$. $e(i,j)$ is the error generated during quantizing the original tone into 0 or 1.

The FGSOI filter is designed to further improve the edge enhancement of the error diffusion proposed by Eschbach et al. [7] while maintaining general information in original image. The function of the FGSOI filter is to differentiate tones of original image in horizontal and vertical directions and fabricate the differentials. To differentiate tones, two Prewitt operator of 3×3 size depicted in Fig. 2 are applied to original image for the corresponding directions. The differentials resulted in for the horizontal and vertical directions are S_i and S_j, respectively, and have near zero for flat area in tone change, positive for decreasing tone, and negative for increasing tone. The values are processed into R_i and R_j to compensate for the different rates of directive bias of error diffusing in the standard error diffusion. R_i and R_j are used to make S_d to suppress excessive edge enhancing in diagonal directions resulted from the summation of them. The results of the intermediate processes are finally combined into $P(i,j)$ and its responses are added to error-diffused image $u(i,j)$.

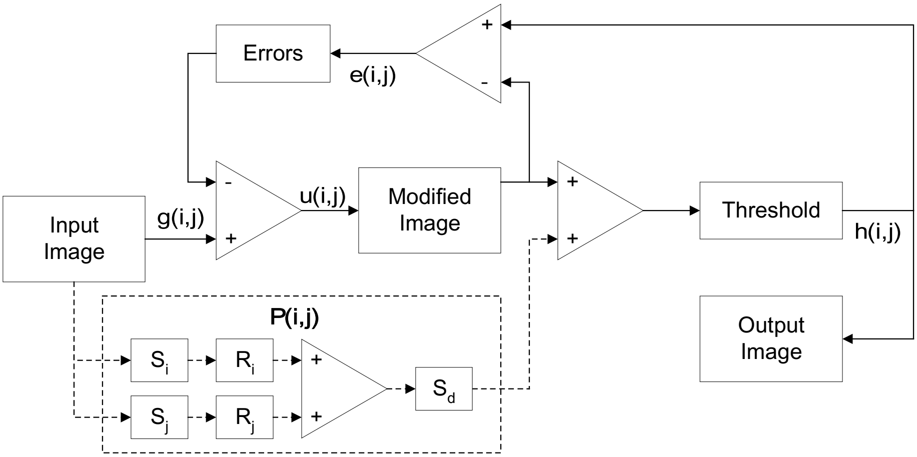


Fig. 1. The edge enhanced error diffusion to which the FGSOI filter is added

-1	0	+1
-1	g(i,j)	+1
-1	0	+1

(a)

-1	-1	-1
0	g(i,j)	0
+1	+1	+1

(b)

Fig. 2. 3×3 differential Prewitt operators for (a) the horizontal direction and (b) the vertical direction

The mathematical expressions involved in each process are presented as follows:

$$S_i = \sum_{l=-1}^{1} g(i+1, j+l) - \sum_{l=-1}^{1} g(i-1, j+l) \cdot \tag{1}$$

$$S_j = \sum_{k=-1}^{1} g(i+k, j+1) - \sum_{k=-1}^{1} g(i+k, j-1) \cdot \tag{2}$$

$$R_i = \frac{a_i}{1 + b_i \cdot |S_i|} \cdot \tag{3}$$

$$R_j = \frac{a_j}{1 + b_j \cdot |S_j|} \cdot \tag{4}$$

$$S_d = \frac{1-\omega}{2} \cdot \cos(4\theta) + \frac{1+\omega}{2}, \quad \theta = \arctan\left(\frac{|S_j \cdot R_j|}{|S_i \cdot R_i|} \right) \cdot \tag{5}$$

$$P(i, j) = \left(S_i \cdot R_i + S_j \cdot R_j \right) \cdot S_d \cdot \tag{6}$$

where, a_i, a_j, b_i and b_j are the coefficients to adjust the effect of edge enhancing and ω is the frequency range in tone change in original image. The combinatorial P shows the similar characteristics to the standard error diffusion for flat area and to the edge enhanced error diffusion proposed by Eschbach et al. for increasing or decreasing area. Such characteristics can supply the ability to improve edge information while retain general information of original image.

3 Evaluation

To evaluate the effect of the proposed edge enhanced FGSOI filter described in Section 2, the two filters of Floyd et al. and Eschbach et al. are compared with the FGSOI filter for Lena image. In this paper three measurement criteria are adopted for the objective comparison of them: radial averaged power spectrum density (RAPSD), edge correlation, and local average accordance. In this section, the three measurement criteria are first described, and then the results of comparison for Lena are presented.

3.1 Radially Averaged Power Spectrum Density for Display Error

The RAPSD is a measurement to determine how similar the original image and the bilevel-toned image are to each other [8]. The preferable bilevel-toned image should not have directive biases in pixel pattern and be radially symmetric. This criterion is tested for power spectrum. The power spectrum is defined as $P(f)$ which conducts two-dimensional Fourier transform on bilevel-toned image, squaring the result, and dividing it by the number of samples. Although $P(f)$ is represented in three-dimension, one-dimensional figure can be presented by the frequency for the easy observation of characteristics. The one-dimensional figure is made by partitioning power spectrum into circular rings of width Δ as shown in Fig. 3.

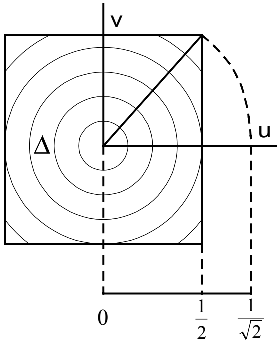


Fig. 3. Partitioning of power spectrum into unit circular rings

When the two-dimensional Fourier transform is designated by $\tau[\cdot]$, the power spectrum density is expressed like this:

$$\hat{P}(u,v) = \frac{1}{I \times J} \left| \tau\left[g(i,j) - h(i,j) \right] \right|^2 . \tag{7}$$

The power spectrum is partitioned into circular rings of the uniform width Δ established in sequence from the center of power spectrum to the outer as seen in Fig. 3. In the figure, it is noted that the circular frequency f_r is distant from the center of circular rings by $\Delta_r / \sqrt{2}$. The RAPSD $P_r(f_r)$ is obtained by integrating the power spectrum over the area of the r-th circular ring and dividing it by the number of samples included in the area as follows:

$$P_r(f_r) = \frac{1}{N_r(f_f)} \sum_{i=1}^{N_r(f_f)} \hat{P}(u,v) . \tag{8}$$

where, $N_r(f_r)$ is the number of samples in the r-th circular ring area.

3.2 Edge Correlation

The most important information of an image is in edge areas. Therefore, it has objectiveness in quality assessment to measure the correlation for edge area between bilevel-toned and original images. The measuring function C for edge correlation is designed as below:

$$D_g(m,n) = g(i,j) - g(i-m, j-n) . \tag{9}$$

$$D_h(m,n) = g_h(i,j) - g_h(i-m, j-n) . \tag{10}$$

$$C=\sum_{i=0}^{I-1}\sum_{j=0}^{J-1}\left(\sum_{m=-1}^{1}\sum_{n=-1}^{1}W_{mn}D_g(m,n)D_h(m,n)\right). \tag{11}$$

where, $g_h(i,j)$ is the continuous-toned image restored from the bilevel-toned image by using a 7×7 low-pass filter designed to consider HVS according to observation distance [9]. W_{ij} is the weighting matrix for the horizontal, vertical and diagonal directions. The rate of the diagonal value to the horizontal and vertical values is $1:\sqrt{2}$ and is normalized such that 0.1465 is obtained for the horizontal and vertical directions, and 0.1035 for the diagonal direction. The finally generated function C evaluates the representing performance for edge area of the bilevel-toned image over the original image. Large C means that edge area of bilevel-toned image is consistent with that of original image.

3.3 Local Average Accordance

The performance how much average tone of local area in original image can be preserved is important as well. This performance is evaluated by a function to measure local average accordance between original image and bilevel-toned one. The original image is divided into rectangles of a specific size and the local average of a rectangle is designated as L_{mg}. The bilevel-toned image is reconstructed by using the 7×7 low pass filter mentioned in Section 3.2 and the local average for a rectangle of the reconstructed image is denoted as L_{mh}. The L_{mg} and L_{mh} are formulated like these:

$$L_{mg}=\frac{1}{M^2}\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}g(i,j). \tag{12}$$

$$L_{mh}=\frac{1}{M^2}\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}g_h(i,j). \tag{13}$$

where, M^2 is the area to get the local averages. The accordance between the two kinds of local average is defined as follows:

$$A_{Lm}=\frac{1}{\frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left(L_{mg}(k,l)-L_{mh}(k,l)\right)^2}. \tag{14}$$

where, N^2 is the number of the local areas. The large A_{Lm} means that local average of the bilevel-toned image is consistent with that of the original image.

3.4 Experimental Results

In this section, the bilevel-toned images, the RAPSDs, the edge correlations, and the local average accordances for each method are presented for Lena image. Display error is defined as the difference between original image and error diffused bilevel-toned image, and the RAPSD for display error will be presented in this evaluation. The experimental results for the filters of Eschbach et al. and the FGSOI are measured for the optimal parameter(s) of their own: 3 is used for the multiplying constant for the filter of Eschbach et al.; and 1.75, 1.75, 0.025 and 0.025 for a_i, a_j, b_i and b_j, respectively, for the FGSOI filter.

The bilevel-toned images generated by the filters of Floyd et al. and Eschbach et al. and the FGSOI are depicted in figures (a), (b), and (c) of Fig. 4, respectively. The figures are cut down from Lena of original size to get better resolution of printed image.

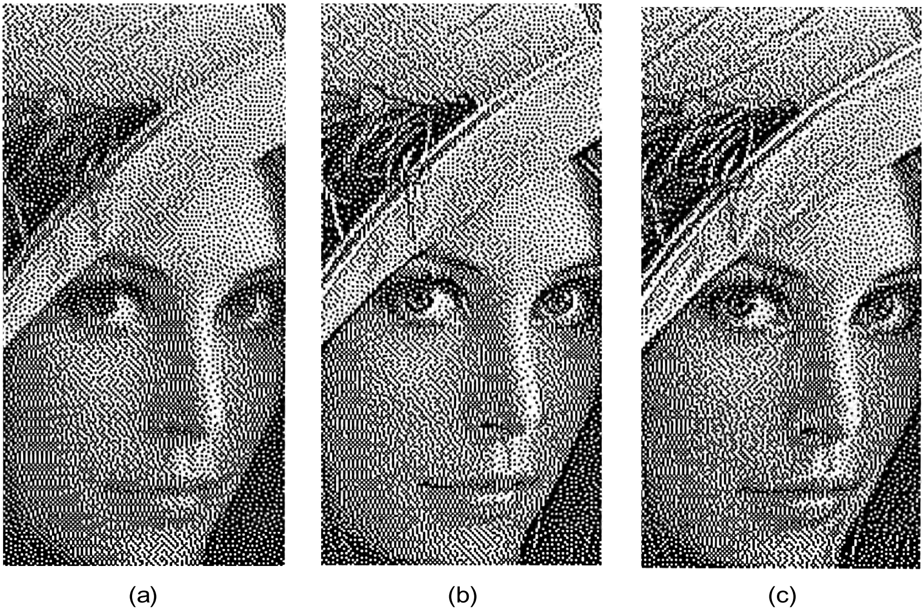


Fig. 4. Bilevel-toned images generated by the filters of (a) Floyd et al. and (b) Eschbach et al. and (c) the FGSOI

The RAPSDs for the display errors made when $\Delta = 0.004$ between the original image and the bilevel-toned images for Lena are displayed in Fig. 5. In figure (a) of Fig. 5, the low frequency range of f_r from 0 to 0.3 generates rare RAPSD and the high frequency range from 0.5 to 0.7 high RAPSD. The RAPSD for the display error by the filter of Eschbach et al. is reported in figure (b) of Fig. 5. The RAPSD for the high frequency range from 0.5 to 0.7 has lower level than that of figure (a) does. Figure (c) of Fig. 5 shows the RAPSD by the FGSOI filter. It is noted that for the whole frequency range, the RAPSD is lower than that of the filter of Eschbach et al., especially for the low frequency range from 0 to 0.3 and for the high frequency rage from 0.5 to 0.7.

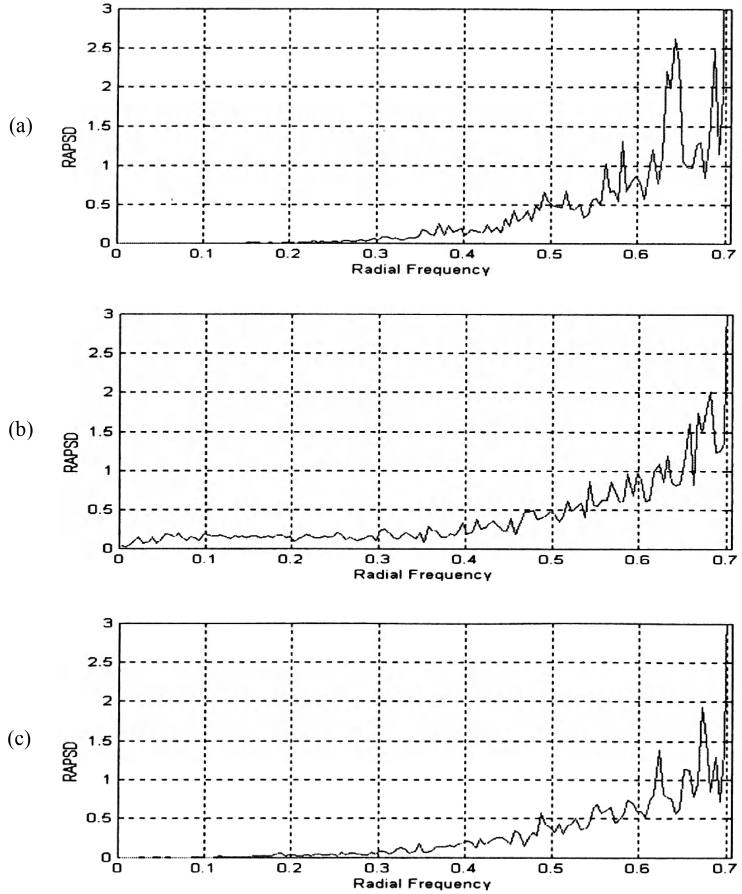


Fig. 5. RAPSD characteristics for the display errors by (a) the filter of Floyd et al.; (b) the filter of Eschbach et al.; (c) the FGSOI filter

The edge correlation and the local average accordance for the bilevel-toned Lena images are recorded in Fig. 6 and Fig. 7, respectively. Fig. 6 presents the edge correlation values as increasing the distances of observation for the three filters. In this figure, the values for the filter of Eschbach et al. and the FGSOI are greater than those for the filter of Floyd et al. The difference between the two groups decreases as observation distance increases, but is recognizable when the bilevel-toned image is observed at a distance of 10 inches, and the values for the FGSOI filter are greater than those of the filter of Eschbach et al. over the distance of 10 inches. Fig. 7 displays the local average accordance values while increasing observation distance for all three filters. In this figure, the values for the filters of Floyd et al. and the FGSOI are even higher than those for the filter of Eschbach et al.

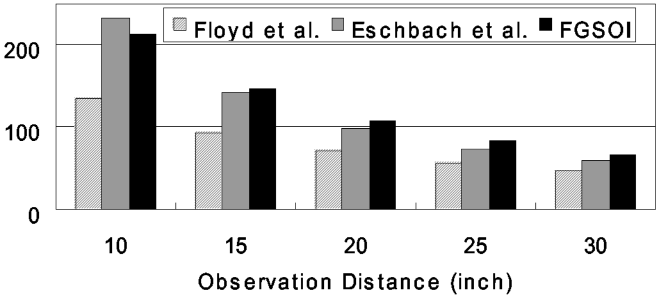


Fig. 6. Comparison of edge correlation values for all the filters

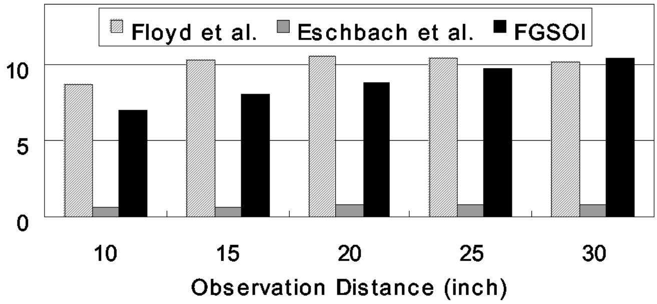


Fig. 7. Comparison of local average accordance values for all the filters

4 Discussion

The outcomes in terms of the visual inspection, the RAPSD for display error, the edge correlation and the local average accordance confirm a distinct improvement of the FGSOI filter compared with the filters of Floyd et al. and Eschbach et al. The filter of Eschbach et al. makes bilevel-toned images sharper than that of Floyd et al. does. However, the filter of Eschbach et al. considers little the fact that the negative effect by the edge enhancing of its own might raise the damage of general information of original images. Compared with the filter of Eschbach et al., the FGSOI filter can enhance further edge information as well as sustain general information.

The comparisons in visual inspection, the RAPSD and the edge correlation convince that the FGSOI filter generates more fine edge information than the filter of Eschbach et al. does without losing general information. It is suggested in Fig. 4 that the filter of Eschbach et al. improve the edge information of Lena over that of the bilevel-toned image by the filter of Floyd et al. and the FGSOI filter further enhance edges especially when the crinkle part of the hat are inspected. Figure (c) of Fig. 5 shows that in the high frequency rage from 0.5 to 0.7 the RAPSD of the FGSOI filter achieves lower level than that of the filter of Eschbach et al. does. It is supported by the outcome in Fig. 6 which all the edge correlation values of the FGSOI over the distance of 10 inches are higher than those of the filter of Eschbach et al. It is because the values of edge correlation present an objective criterion about how much edge information of original image is preserved in bilevel-toned image as described in Section 3.2.

The negative effect of the filter of Eschbach et al. can be found out to compare the RAPSD in the low frequency rage from 0 to 0.2 of figure (b) of Fig. 5 with that of figure (c). It becomes clear when the local average accordance values made by the filters of Eschbach et al. and the FGSOI are examined. As seen in Fig. 7, the distortion of the bilevel-toned image in general information was made seriously by the filter of Eschbach et al. Putting together the experimental evidences until now, it can be argued that the FGSOI filter conducts more effectively the edge enhanced error diffusion than the filter of Eschbach et al. does.

5 Conclusion

So far in this paper we have investigated the FGSOI filter-based error diffusion to more emphasize edge information of original image while retaining general information. The error diffusion of the filter was applied to Lena image and the resulting bilevel-toned image was analyzed for the objective criteria in visual inspection, the RAPSD for display error, the edge correlation, and the local average accordance. From the experimental results, it can be finally concluded that the FGSOI filter presents superior properties than the filters of Floyd et al. and Eschbach et al. do for the low frequency range that includes general information and for the high frequency range that includes most edge information in original image.

References

1. Floyd, R. W., Steinberg, L.: An Adaptive Algorithm for Spatial Greyscale. SID **17** (1976) 75-77
2. Counse, K. R., Roska, T., Chuam L. O.: Image Halftoning with Cellular Neural Networks. IEEE Trans. Circuits and Systems-II **40** (1992) 267-283
3. Jarvis, J., Judice, C., Ninke, W.: A Survey of Techniques for Display of Continuous-Tone Pictures on Bilevel Displays. Comp. Graph. Image Processing **5** (1976) 13-40
4. Wong, P. W.: Adaptive Error Diffusion and Its Application in Multiresolution Rendering. IEEE Trans. Image Processing **5** (1996) 1184-1196
5. Sullivan, J., Miller, R., Pios, G.: Image Halftoning Using a Visual Model in Error Diffusion. J. Opt. Soc. Am. A. **10** (1993) 1714-1724
6. Pappas, T. N., Dong, C. K., Neuhoff, D. L.: Measurement of Printer Parameters for Model-Based Halftoning. Journal of Electronic Imaging **2** (1993) 193-204
7. Eschbach, R., Knox, K.: Error Diffusion Algorithm with Edge Enhancement. J. Opt. Soc. Am. A. **8** (1991) 1884-1850
8. Lau, D. L., Arce, G. R., Gallagher, N. C.: Green-Noise Digital Halftoning. Proceedings of IEE **86** (1998) 2424-2444
9. Pappas, T. N., Neuhoff, D. L.: Least-Squares Model-Based Halftoning. IEEE Trans. on Image Processing **8** (1999) 1102-1116

Capitalizing Software Development Skills Using CBR: The CIAO-SI System

Roger Nkambou

Department of Computer Science
University of Quebec at Montreal
nkambou.roger@uqam.ca

Abstract. This paper presents a CBR-Based tool dedicated to the capitalization of development experience/knowledge within a software development company. The tool (called CIAO-SI) provides developers with appropriate assistance during the development process. The core of CIAO-SI is a memory of software artifacts (models, documents, source code…) which may be produced in a given CASE tool and easily retrieved using a relevant indexation structure. The retrieval process is supported by domain and tasks ontologies related either to some software development processes (RUP, Merise…) or to some application domains (Human Resources Management, accounting, banking…).

1 Introduction

Experience capitalization has become one of the major challenging issues in software engineering in general and software development in particular. The main idea is to take advantage of the experience cumulated throughout years when addressing new software development projects. Software reuse is one of the major issues when capitalizing experience. It is widely accepted by authors and professionals as a mean of capitalizing know and know-how gained during previous development projects. Knowledge management can contribute to this effort and is now considered as a potential source of productivity and quality improvement in software development [1]. Exploiting software development experience in terms of knowledge related to development process (meta-knowledge [2]) and artifacts (knowledge embedded in products knowledge [2]) is in our own opinion, a powerful means that can enable very significant software productivity, quality and costs improvements. This ability is not available till nowadays in major software development CASE tools found on the market (Rational Rose, Objecteering, Together…).

Though these CASE tools do help produce software artifacts during the development process, nothing is done to manage a memory of these artifacts for future development projects. The idea in the CIAO-SI project is to take advantage of the CBR methodology/technique [3]), and manage such a memory, facilitating reuse in development activities. Furthermore, the CIAO-SI project is interested in the development of a system that can assist developers during software development, reducing training costs (especially for new developers). Finally the CIAO-SI project addresses the problem of poor documentation during software development projects which is a real and serious problem in many development companies. The CIAO-SI project was ini-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 483-491, 2004.

tiated in 2001 by Roger Nkambou, professor of computer science and Director of the *GDAC* laboratory in UQAM, and Raphael Mbogni, president of *Le Groupe Infotel Inc*. This company is specialized in software development and as such, has developed a great expertise regarding software development. Thus, it offers a good framework for testing the CIAO-SI system.

The purpose of this paper is to present the first results of the CIAO-SI project relatively to all the issues listed above. A brief review of tools and research work related to knowledge management in software engineering is proposed in section 2. An architecture and functionalities of our CBR based tool (the CIAO-SI tool) for software development experience capitalization is presented in section 3. We present an example of how to use the system in section 4 and conclude.

2 Related Works

Software engineering is a knowledge intensive discipline and as such, could benefit from research on knowledge management. There are many research works and tools (research prototypes or commercial tools) dedicated to knowledge management (KM) in software engineering activities [4, 5, 6, 7, 8]. These tools often take into account experience/knowledge from previous projects. Experience/knowledge Software engineering in the field of includes artifacts related knowledge (requirements, design, and other software documentation/models/data) as well as process related knowledge (design method tools, software best practices, technologies, lessons learned, various models, and data) [6]. We have examined some tools that offer appropriate support for knowledge management in software engineering.

Rebuilder [8] is one of the first tools we examined. It is a CASE (Computer Aided Soft-ware Engineering) tool, which applies analogical reasoning to the reuse and design of object-oriented software. In ReBuilder, only UML class diagrams are used for reasoning.

Design Rationale is another tool we examined. It is a framework for creating organizational memory that attempts to preserve information about the development of a software product [4]. It addresses the organization of the software development memory. The idea behind Design Rationale is that during a software development project, different solutions to problems are tested and many decisions are taken based on the results of these tests [4, 9]. Design Rationale has concentrated effort on knowledge *capture* and *representation*. Capturing, recording or representing decision is a particularly difficult problem [12]. Recording all decisions made, as well as those rejected, can be time consuming and expensive. The more intrusive the capture process is, the more designer resistance will be encountered. The main limitation is the fact that no explanation is given on the decision-making process. Design Rationale also acknowledges the need for capturing information about solutions that were considered, but not implemented.

Tools like *Kibitzer* [10] and *RFML Workbench* [11] are among few knowledge acquisition tools specially targeted to aiding domain experts in the identification/conceptualization phase of the knowledge-based system development cycle. They can interactively acquire knowledge about the domain (domain object, concepts and corresponding relationships) for which the software is to be developed [4, 10, 13]. Kibitzer has contributed efficiently to identify various types of knowledge required for domain modeling. Thus, the model representation language resulting from this analysis is relatively simplistic; it is useful for expressing the static aspect of a domain (object) but not its dynamic concept (process, action) [13]. The validity and relevance of the domain model depend on the user knowledge about the software domain (it may contain incoherencies or redundancies).

Tools such as *KBRAS* [14], *CAESAR* [15], address particular phases of the software engineering process. KBRAS supports reusability of software components by classifying and retrieving requirements for other software applications. CAESAR is a case-based reasoning system for constructing programs, in a specific domain, from existing program components. CAESAR helps users building a first draft of new a program from the given specification, using library programs of an existing software. The main constraint for these tools is that the user must be a domain expert, in order to understand how the library programs must be decomposed in terms of their functional specifications.

Grupe et al. [7] have proposed a prototype that applies CBR to the software development process. In their prototype, a case is limited to external aspects of a software project (project name and description, hardware environment in which the project was constructed, language or package under which the project was developed, number of function points…). The reuse is limited to artifacts related knowledge.

Klaus [5], introduced a few years ago a tool architecture and its underlying methodology for reuse and continuous learning of all kinds of software engineering experience. The architecture extends existing ones [6] by supporting the whole process of reuse and learning for arbitrary experience items (artifacts). Although this architecture does contribute to extend reusability concept to all the development phases, it does not leave much room for the semantic dependences between these artifacts (the experience base is limited to a set of artifacts). The level of reusability (reuse granularity) is limited to artifacts, which implies a total overlook of the solution. Moreover, the production, the completeness and the format of the artifacts are not supported by a well-defined methodology; it depends on the experience of the project team.

The last relevant tools we examined are Dj□□vu and RSL. *Déjà Vu* [16] is a CBR system for code reuse and generation using an hierarchical CBR. In this tool, the problem space is partitioned into sub-cases (abstract and concrete). Each case has a description part (specification part) and a solution part (program). One of the limitations of this tool is that reusability remains at code level; moreover, search for similar cases is focused on problem decomposition rather than problem-space decomposition.

As for *RSL* [17], it is a system that allows the reuse of code and design knowledge. Component retrieval is done using natural-language queries, or using specific attributes. Component ranking is an interactive and iterative process between RSL and the user. Like other tools listed above, the reuse is limited to artifacts related knowledge and nothing is said about the case structure, the completeness, the consistency and relevance of the information stored in the case.

From all that precedes, it appears that all examined tools focus on specific development phases and/or are not supported by any development process. Moreover, many of them if not all do not offer a personalized assistance to the developer. These limitations are taken into account in the CIAO-SI project.

3 The CIAO-SI Tool

3.1 System Functionalities

Software artifacts memory and reuse. The CIAO-SI tool manages a memory of artifacts produced during software development (documentation, planning, source code, diagrams…). The system helps in producing or modifying these artifacts by invoking appropriate development tools. For new developments, artifacts are not produced from scratch: The system offers the possibility to reuse and adapt already stored artifacts. Details concerning cases structures and memory indexation strategy can be found in [15, 16].

Application domain knowledge memory. The CIAO-SI tool manages a memory of knowledge related to various application domains in terms of domain application ontologies (domain ontology and tasks ontology). These ontologies are very useful for a better characterization and classification of software development projects. For each application domain, a set of concepts and tasks related to the domain is stored in the application domain knowledge memory. In a case based reasoning perspective, they will be used to refine the similarity between projects. For *Le Groupe Infotel Inc*. case, we have identified four main application domains: Banking management, accounting management, human resources management and payroll management. We are currently building ontologies for human resources management.

Process knowledge memory. The CIAO-SI tool manages a memory of knowledge related to development process in terms of process ontologies (domain ontology and tasks ontology). These ontologies are very useful for a better follow-up and tracking of development activities and software artifacts related to each activity. Rational Unified Process (RUP) is one of the processes the system will be working with. Thus, we've developed and stored in the process knowledge memory, a domain ontology and a task ontology related to this process. We are expecting to do the same work very soon with the *Merise* development process.

Assistance to developers. The CIAO-SI tool offers assistance to developers especially for solution adaptation activities: it is useful for the quality of artifacts resulting from the adaptation process, preventing inconsistencies. It also facilitates the training of developers to development processes by guiding them during the adaptation process (ensuring that the phases of the development process are respected and all artifacts for each phase are produced). Process knowledge memory is used here. E-learning techniques can also be put to contribution in adapting the assistance to the developer (developer model, tutorial strategies [tutoring, coaching, criticizing…]).

3.2 System Architecture

From Fig. 1, it can be seen that the CIAO-SI tool consists of three main components: the expert workspace, the developer workspace and the management component.

The developer workspace. This component allows a software engineer to design and build a new application from existing "similar" ones. "Similar" applications are retrieved from the *case base* via the *case retrieval* module based on a certain number of *retrieval criteria* (characteristics of the new application, chosen by the developer from a given *criteria set*). In order to reduce the number of *retrieved cases*, the retrieval algorithm uses concepts and tasks from the considered application domain ontologies. These concepts and tasks are stored in the *Application Domain Knowledge Base* (*ADKB*) and managed by the *Application Domain Knowledge Manager* (*ADKM*). Depending on the case description and the score, one or many cases are retrieved from the case base. Retrieved cases are weighted according to their similarity to the case problem (new application); in the next section, we present some details of the retrieval process. The developer chooses one and adapts it using appropriate *CASE Tools,* in interaction with the *Assistant manager*. The adaptation is supported by the *Process Knowledge Base* (*PKB*) containing concepts, tasks and best practices related to development processes (RUP, Merise…). The Assistant Manager also offers to the developer a certain number of ready-to-use templates that present documents and artifacts. In order to personalize the assistance, the Assistant Manager uses the developer profile stored in the *Profile Database (PDB)* and managed by the *Profile Manager (PM*). The adapted case or *candidate case* that is the new application is stored in the *buffer* (temporary case base used for storing not yet validated cases). The validation is performed in the Expert workspace.

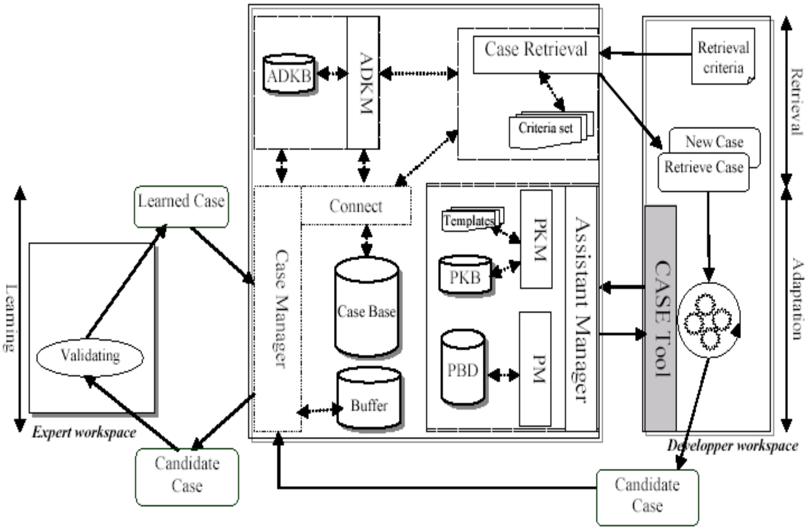


Fig. 1. CIAO-SI system architecture

The expert workspace. This component allows an expert (someone who has an expertise in an application domain) to validate cases contained in the buffer, in order to: eliminate inconsistencies, ensure that the case is relevant according to the application domain, and ensure that the case is not redundant in the case base. The validated case or *learned case* is stored in the *case base* managed by the *Case Manager*. It should to be noted that a case must always be validated before it is stored in the *case base*.

The management component. This component is the core of the system. It comprises four main management modules: the *Case Retrieval Manager*, the *Application Domain Knowledge Manager*, the *Assistance Manager* and the *Case Manager.* The *Case Retrieval Manager* is dedicated to the retrieval of a case or set of cases that match best with a new application to be developed, given its characteristics. The *Application Domain Knowledge Manager* is used for the management of all the knowledge related to application domains. As for the *Assistance Manager,* it manages all the knowledge related to development processes and ensures the follow-up of developers during the adaptation of a case. Finally the *Case manager* carries out the basic operations on cases (add, modify and delete).

3.3 The Retrieval Process in CIAO-SI

The retrieval process is based on an indexation structure, which includes two parts: a dynamic inductive tree and an inverted list**.**

Dynamic inductive tree: this sub-structure is built dynamically according to a set of weighted retrieval criteria. The order in which the criterion are evaluated (succession of nodes) is determined by the indexation algorithm during execution, which makes it possible to take into account the criteria according to their relevance (weight) as defined by the user. Cases are classified in the tree leaves by group of satisfaction (set of cases). The use of the dynamic inductive tree is very significant at this level because it contributes to strongly partition the data into sets of cases with increasing satisfaction level.

Inverted List: This sub-structure is similar to the one used for indexing documents with keywords in documentary databases: In documentary databases, users are interested in obtaining the list of documents containing a set of keywords, while in our case, we are interested in determining the list of cases containing a set of concepts. References to selected concepts of the considered application domain are stored in the sub-structure (for some application domains, a domain ontology is available in the knowledge base). The user chooses the concepts from a list of concepts found in the domain. A weight is affected to each concept referenced in the sub-structure, and references to cases containing the concept are also kept. The weights and the number of references to concepts for each case are used in combination to sort the cases. It should be noted that cases considered here are the one appearing in the leaves of the inductive tree presented above. The retrieval algorithm takes advantage of this structure. The retrieval process has two steps.

The first step uses a set of weighted retrieval criteria to locate candidate cases. This set may contain more than one category of retrieval criteria. Selected criteria are used to express constraints to be achieved by candidate cases. A dynamic inductive tree is built based on these criteria. The dynamic inductive tree is used to partition the case base into many subsets of cases. Each subset is made of cases that have the same similarity level according to the selected criteria. The computation of similarity level is done using Riesbeck's formula [18]. Candidate cases are extracted from these subsets. A candidate case is one belonging to a subset with a similarity level higher than a threshold set by the user.

The second step of the retrieval process builds an inverted list of concepts from candidate cases selected by the user. The inverted list consists of concepts and for each concept, references to the case(s) containing the concept. On another hand, a list of available concepts of the considered domain is extracted from the application domain ontology. The user selects appropriate concepts according to the application to be built. Concepts of the inverted list are matched with concepts selected and weighted by the user. All cases referenced in the inverted list are ranked according to the number and weight of concepts (those selected and weighted by the user) that they contain. These cases are proposed to the user as final selected cases.

The proposed retrieval techniques has been successfully used and is better than most existing techniques in terms of the relevancy of the selected cases. Other details on this technique and its evaluation can be found in [17].

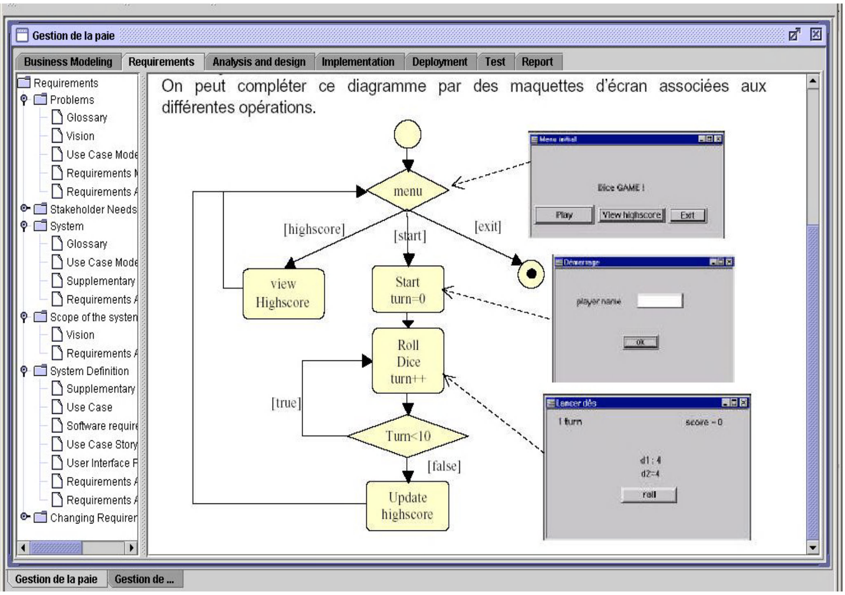
4 Using CIAO-SI : An Example

Let us suppose that a software engineer wants to develop a new banking application, using the CIAO-SI tool. He logs into the tool as a developer and the developer workspace/interface is displayed. Then he clicks on the *new application* button. He is prompted with a window presenting a list of different application domains for which ontologies are available in the system. From this list, he selects the *Banking* application domain and continues the process. He is then prompted with a window displaying different categories of software characteristics. These characteristics are used as retrieval criteria.

For each category of retrieval criteria (general, architecture, quality…), he selects and instantiates the criteria that fit with the application to be developed. To each selected criterion, a weight is affected indicating the importance of the criterion for the developer. After selecting and instantiating the retrieval criteria, the developer is prompted with a window presenting all the cases pre-selected from the case base, that match best with the application to be developed, according to the selected criteria values and weights. This selection is computed using the induction tree as stated in section 3.3.

For each pre-selected case, the “similarity” level (similarity with the application to be developed) in percentage is given. Details of the application corresponding to each the pre-selected case are also provided. Based on these details and the “similarity” levels, the developer selects the cases he considers to be close to the application he wants to develop. At this stage, the number of cases to be examined can still be high. In order to reduce this number, a second level of case selection is performed on the pre-selected cases using ontologies (domain ontology and tasks ontology) related to the *Banking* application domain (the application domain of the application to be developed).

All concepts and tasks considered relevant for the application to be developed are selected by the developer. Selected concepts and tasks are used by the retrieval process, which goes trough the inverted list and comes out with appropriated cases. These cases are then loaded in the main window of the developer workspace/interface (figure 2).



On peut compléter ce diagramme par des maquettes d'écran associées aux différentes opérations.

Fig. 2. A selected case in the developer workspace of the CIAO-IS tool

The different phases of Rational Unified Process (RUP) appear at the top of the screen (Figure 2). The left side of the screen contains the list of artifacts for a selected phase and a selected application (the list of applications appears at the bottom of the screen). Finally the right side of the screen contains details of a selected artifact. For each selected case, the developer has all the artifacts (reports and models) produced during the development of the corresponding application. He just has to adapt them for his new application. During the adaptation phase, he can interact with the Assistant manager for advises and validations.

5 Conclusion

We have presented a comprehensive architecture and functionalities of a CBR based tool for software development experience capitalization. The main advantage of the presented tool is the scope of reuse supported. While existing tool focus on specific development phases and are not supported by any development process, our tool covers all the development life cycle and is supported by a development process (RUP for instance). The CIAO-SI tool also provides personalized assistance to developers on the software application domain and the development process, during the adaptation of a project retrieved from the base. It allows developers to continue using their favorite CASE tool, as long as it is XMI compliant. Future research works will include the strengthening of the case structure, the refinement of the indexation technique and the production of the first complete prototype.

Acknowledgement. We would like to thank **Le Groupe Infotel Inc.** for their financial support to this work.

References

[1] Torgeir Dingsøyr, *Knowledge Management in Medium-Sized Software Consulting Companies*, Doktor Ingeniør thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, 2002. ISBN 82-471-5401-3, pp. 256.

[2] Engelhart, P., M., *Knowledge Management in Software Engineering: A State-of-the-Art-Report*, 29 November 2001, Air Force Research Laboratory, Information Directorate/IFED, 32 Brooks Road, Rome, NY 13441-4505.

[3] A.Aamodt and E.plaza, *Case-Based Reasoning: fundational issues, methodological variations, and system approaches*, Aicom 7(1) 39-59 Mar 1994.

[4] Ioana Rus, Mikael Lindvall, and Sachin Suman Sinha, *Knowledge Management in Software engineering: A State-of-the-Art-Report,* A Data & Analysis Center for Software Task Contract No. SPO700-98-D-4000, Center for Experimental Software Engineering Maryland and The University of Maryland.

[5] Klaus-Dieter Althoff, Andreas Birk, Susanne Hartkopf, Wolfgang Müller, Markus Nick, Dagmar Surmann, Carsten Tautz, *Managing Software Engineering Experience for Comprehensive Reuse* SEKE'99, Kaiserslautern, Germany, June 1999

[6] Klaus-Dieter Althoff, Markus Nick, Carsten Tautz *An Application Implementing Reuse Concepts of the Experience Factory for the Transfer of CBR System Know-How*, Fraunhofer Institute for Experimental Software Engineering. CBR-PEB, Sauerwiesen 6, D-67661 Kaiserslautern, Germany.

[7] Fritz H. Grupe, Robert Urwiler, Narender K. Ramarapu and Mehdi Owrang. *The application of case-based reasoning to the software development process* Information and Software Technology, Volume 40, Issue 9, 15 September 1998, Pages 493-499.

[8] Paulo Gomes, Francisco C. Pereira, Paulo Paiva, Nuno Seco, Paulo Carreiro, José L. Ferreira, Carlos Bento, *Case Retrieval of Software Designs using WordNet*, In Proceedings of the European Conference on Artificial Intelligence 2002 (ECAI'02).

[9] Allen Dutoit, *Design Rationale: Concepts, Technique, and Use* http://www.globalse.org/teaching/ss98/dr/slides/Introduction.pdf.

[10] Schoen, E., *Intelligent Assistance for the Design of Knowledge-based Systems*, Ph.D. Thesis, Stanford University. (1991).

[11] Gibson, M. and Kevin, C., *Domain Knowledge Reuse During Requirements engineering*, in 7th International Conference on Advanced Information Systems Engineering, CAISE, 1995, pp. 283-296.

[12] J. Burge, D. C. Brown, *Reasoning with Design Rationale*, in Design Research Group Department of Computer Science http://www.cs.wpi.edu/~dcb/Papers/AID00-janet.pdf.

[13] Byung Joon Park, Domain Modeling For Knowledge-Based Systems, Thesis, University of Illinois at Urbana-Champaign, 1997.

[14] K. Zeroual, P.N. Robillard, KBMS: A Knowledge-based System for Modeling Software System Specifications, IEEE Trans. on Knowledge and Data Engineering, vol. 4, ndeg. 3, 1992.

[15] G. Ngantchaha, R. Nkambou, V. Bevo Software engineering knowledge capitalization *using CBR: case structure. Accepted for the AIA 2004 conference.*

[16] Donfack, H., Nkambou, R., Bevo, V. Case Retrieval in Software using inductive tree and inverted lists. *Accepted for the AIA 2004 conference.*

[17] Donfack, H. An indexing technique for CBR retrieval process. Master thesis. Department of computer science. University on Quebec at Montreal. Canada. 2004.

[18] Riesbeck, C. and Schank, R. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates. Hillsdale, NJ, USA. 1989.

A Hybrid Case Based Reasoning Approach for Monitoring Water Quality

Claudio A. Policastro, André C.P.L.F. Carvalho, and Alexandre C.B Delbem

Institute of Mathematics and Computer Sciences -- University of Sao Paulo.
Av. Trabalhador Sao-Carlense, 400 -- 13560-970 -- Sao Carlos, Sao Paulo, Brazil.
{capoli,andre,acbd}@icmc.usp.br

Abstract. There is a growing concern with the impact of human intervention in the environment. The impact on the environment of toxic waste, from a wide variety of manufacturing processes, is well known. More recently, however, it has become clear that the more subtle effects of nutrient level and chemical balance changes arising from farming land run-off and sewage water treatment also have a serious, but indirect, effect on the states of rivers, lakes and even the sea. This paper investigates the use of a hybrid Case Based Reasoning system for monitoring water quality based on chemical parameters and algae population.

1 Introduction

There is a growing concern with the impact of human intervention in the environment. The impact on the environment of toxic waste, from a wide variety of manufacturing processes, is well known. More recently, however, it has become clear that the more subtle effects of nutrient level and chemical balance changes arising from farming land run-off and sewage water treatment also have a serious, but indirect, effect on the states of rivers, lakes and even the sea [4].

This paper investigates how Machine Learning techniques can be employed to support environmental control, by monitoring water quality based on chemical parameters and algae population. For such, this work uses a dataset composed by samples taken from sites on different European rivers over a period of approximately one year. These samples were analyzed for various chemical substances including: nitrogen in the form of nitrates, nitrites and ammonia, phosphate, pH, oxygen, chloride. In parallel, algae samples were collected to determine the algae population distributions. It is well known that the dynamics of the algae community is determined by external chemical environment with one or more factors being predominant. While the chemical analysis is cheap and easily automated, the biological part involves microscopic examination, requires trained manpower and is therefore both expensive and slow [4].

The relationship between the chemical and biological features is complex and demands the application of advanced techniques. This paper investigates the use of a Case Based Reasoning (CBR) system for monitoring water quality using chemical indicators and algae population.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 492-501, 2004.

This paper is organized as follows: Section 2 briefly introduces the CBR paradigm. Section 3 discusses some previous works. Section 4 presents the hybrid system architecture. Section 5 shows experimental results. Section 6 presents the final considerations.

2 Case Based Reasoning

CBR is a methodology for problem solving based on past experiences. This technique tries to solve a new problem by employing a process of retrieval and adaptation of previously known solutions of similar problems. CBR systems are usually described by a reasoning cycle (also named CBR CYCLE), which has four main phases [1]:

1. *Retrieval*: according to a new problem provided by the user, the CBR system retrieves, from a Case Base (CB), previous cases that are similar to the new problem;
2. *Reuse*: the CBR system adapts a solution from a retrieved case to fit the requirements of the new problem. This phase is also named *case adaptation*;
3. *Revision*: the CBR system revises the solution generated by the *reuse* phase;
4. *Retention*: the CBR system may learn the new case by its incorporation in the CB, which is named *case learning*. The fourth phase can be divided into the following procedures: relevant *information selection* to create a new case, *index composition* for this case and *case incorporation* into the CB.

CBR is not a technology developed for specific purposes, it is a general methodology of reasoning and learning [1], [7], [22], differing in important aspects from other AI paradigms [1]:

- CBR can use specific knowledge from previous problems;
- The reasoning from previous problems is a powerful strategy for problem solving, which is frequently applied by human beings;
- The CBR paradigm is based on psychological evidence.

The CBR paradigm is supported by two main principles [9]. The first principle says that the world is regular: similar problems have similar solutions. Consequently, the solutions of similar problems are a good starting point for the solution of new problems. The second principle states that problems tend to repeat. Thus, new problems tend to be similar to previous problems.

3 Previous Works on Hybrid Case Reasoning Systems

CBR and Artificial Neural Networks (ANN) have been successfully combined, particularly in situations requiring a functional integration [11], [12], [18]. Several approaches for integration of CBR and ANN employs neural networks in particular stages of the CBR CYCLE, specially, in the case retrieval.

In [11], an incremental ANN is used for case learning and retrieval. Three layers compose the network: the input layer, whose units receive the problem attributes, the output layer, which contains a neuron for each case, and the internal layer, where each neuron represents a cluster of similar cases (also named prototype). The main idea is the construction of a simple system of case organization with two levels of memory: one containing prototypes of cases and another comprising instances of real cases.

Corchado et al. [3] use an ANN based on *Radial Basis Function* (RBF) model [14] for case adaptation. When a new problem is presented to the CBR system, a set of similar previous cases is retrieved. These cases are then used to train the RBF network. After the training, the new problem description is presented to the network, which works as a function mapping the current problem to a solution.

4 Hybrid System Architecture

This work investigates the use of committees of *Machine Learning* (ML) algorithms to perform the adaptation of cases retrieved from a CB (second phase of the CBR CYCLE) in a Case Based Reasoning system for monitoring water quality. The committees investigated are composed of ML algorithms, here named estimators, based on different paradigms. One ML algorithm, here named combiner, combines the outputs of the individual estimators to produce the output of the committee. The estimators and the combiner are used to perform adaptations in the recovered solution to predict water quality. Similar approaches are presented in [15], [16]. The following ML algorithms compose the committee:

- Estimators: a *Multi Layer Perceptron* (MLP) neural network [6]; a symbolic learning algorithm M5 [21]; a *Support Vector Machine* (SVM) technique [20]
- Combiner: in this work were investigated three ML algorithms as the combiner of the committee: a MLP neural network, the M5 learning algorithm and the SVM technique. The combiner receives the outputs from the other three algorithms as input, combines the results, and produces the output of the committee.

MLP networks are the most commonly used ANN model for pattern recognition. A MLP network usually presents one or more hidden layers with nonlinear activation functions (generally sigmoidal) that carry out successive nonlinear transformations on the input patterns [6].

M5 is a learning algorithm that generates models on the form of regression trees combined with regression equations (Model Tree) [21]. This model works similarly to a classification tree. However, the leaves contain linear expressions instead of predicted values. The Model Tree is constructed by a divide-and-conquer approach that recursively creates new nodes. This approach applies a standard deviation test to divide the remaining data into subsets and associates the test results to each new node. This process is carried out for all data subsets, creating an initial model. Afterward, a linear model is calculated for each inner node of the tree using a standard regression process. Next, the tree is pruned by evaluating the linear model of each node and its sub-trees [17].

SVM is a family of learning algorithms based on the statistical learning theory [20]. It combines generalization control with a technique that deals with the dimensionality problem[1] [20]. By using hyperplanes as decision surface, it maximizes the separation borders between positive and negative classes. In order to achieve these large margins, SVM follows a statistical principle named *structural risk minimization* [20]. Another central idea related to SVM algorithms is the use of kernels to build support vectors from the training dataset.

4.1 Case Adaptation

The approach for case adaptation employs two modules. The first module (adaptation pattern generation) produces a dataset of adaptation patterns. This dataset is then used by the second module (case adaptation mechanism). The second module trains a committee of ML algorithms to automatically perform case adaptation. This approach extends the approach proposed in [15], [16] by exploring the use of a hybrid committee of ML algorithms as case adaptation mechanism. This approach assumes that a CB is representative [19], i.e. the CB is a good representative sample of the target problem space. Therefore, no re-training of the adaptation mechanism is required when the system creates new cases during the reasoning process.

4.1.1 Adaptation Pattern Generation

The dataset generation module is capable of extracting implicit knowledge from a CB. This module employs an algorithm that is similar to that proposed in [15],[16] (see Algorithm 1).

Algorithm 1

```
function AdaptationPatternGenerate (CasesNumber, Component)
  for all cases from the original case base
    ProofCase ∧ ProofCaseExtract ()
    ProofDescrpt ∧ DescriptionExtract (ProofCase)
    ProofSolution ∧ SolutionExtract (ProofCase, Component)
    RetrievedCases ∧ Retrieve (ProofDescrpt, CasesNumber)
    for all RetrievedCases
      RetDescrpt ∧ DescriptionExtract (RetrievedCases(i))
      RetSolution ∧ SolutionExtract (RetrievedCases(i),
                                      Component)
      MakeAdaptationPattern(ProofDescrpt, RetDescrpt,
                            RetSolution, ProofSolution)
     end for
  end for
end function
```

[1] Machine Learning algorithms can obtain a poor performance when working on data sets with a high number of attributes. Techniques of attribute selection can reduce the dimensionality of the original data set. SVM is a ML Algorithm capable of keeping a good generalization even for data sets with many attributes.

Initially, the pattern generation algorithm extracts a case from the original CB and uses it as a new problem (*ProofCase*) to be presented to the CBR system. The remaining cases compose a new CB without the proof case. Next, the algorithm extracts, from the proof case, the attributes of the problem (*ProofDescrpt*) and a component (indicated by *Component*) of the solution (*ProofSolution*). Later, the algorithm returns the *N* most similar cases from the *ProofDescrpt* (*RetrievedCases*), where *N* is a predefined value. For each retrieved case, the attributes of the problem (*RetDescrpt*) and a component of the corresponding solution (indicated by *Component*) are extracted (*RetSolution*). Next, the algorithm generates the adaptation patterns using as input attributes: the problem description stored in the proof case, the problem description stored in the retrieved case, a component solution stored in the retrieved case; and as output attribute: a solution component stored in the proof case. Finally, the generated datasets are used to train the committee of ML algorithms. First, the MLP, the SVM and the M5 algorithm are trained individually using the adaptation pattern dataset generated. Next, the output of these three ML algorithms are combined to produce a training dataset for the combiner of the committee (MLP or M5 or SVM). The general structure of the adaptation pattern generation is shown in the Figure 1:

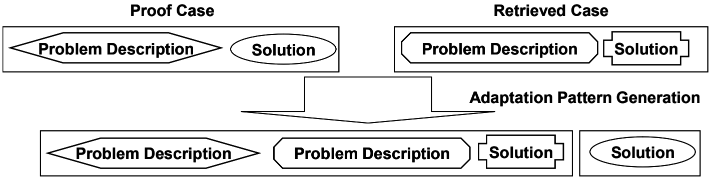


Fig. 1. Adaptation pattern structure.

4.1.2 Case Adaptation Mechanism

The case adaptation mechanism allows the learning of the modifications that need to be performed in the components values of the retrieved solutions in order to achieve an adequate solution for a new problem. The most important characteristic of this mechanism is the employment of implicit knowledge obtained from the CB with a minimum effort for the knowledge acquisition. The case adaptation process is shown in the Algorithm 2.

```
Algorithm 2
function Adaptation (Description, RetrievedCase, Component)
   RetDescription ∧ DescriptionExtract RetrievedCase)
   RetSolution ∧ SolutionExtract (RetrievedCase, Component)
   InputPattern ∧ MakeInputPattern (Description,
                                    RetDescription,
                                    RetSolution)
   Acts ∧ AdaptationMechanism (Normalization(InputPattern),
                               Component)
   NewSolution ∧ ApplyActs (RetSolution, Acts, Component)
   return NewSolution
end function
```

When a new problem is presented to the CBR system, the most similar case store in the CB is obtained by a retrieval mechanism [5], [10]. This case (*RetrievedCase*) is sent to the adaptation mechanism together with the problem description (*Description*). The adaptation algorithm, in turn, extracts the attributes from the new problem (*RetDescription*). Next, for each component of the retrieved solution (indicated by *Component*), the algorithm extracts the corresponding solution and generates an adequate input pattern for the committee of ML algorithms developed for this component. Afterwards, the committee indicates the required modifications in the component of the retrieved solution (*Acts*). Finally, these modifications are applied to the current component in order to obtain the solution for the new problem (*NewSolution*).

The case adaptation approach works only with a single component of the solution of a case. This approach can be easily extended for domains where the solution of the cases has more than one component, by treating each solution component as a distinct problem. This strategy keeps this approach independent of the structure of the case solution. Figure 3 shows the general architecture of the hybrid approach, highlighting the investigated algorithms in the CBR CYCLE.

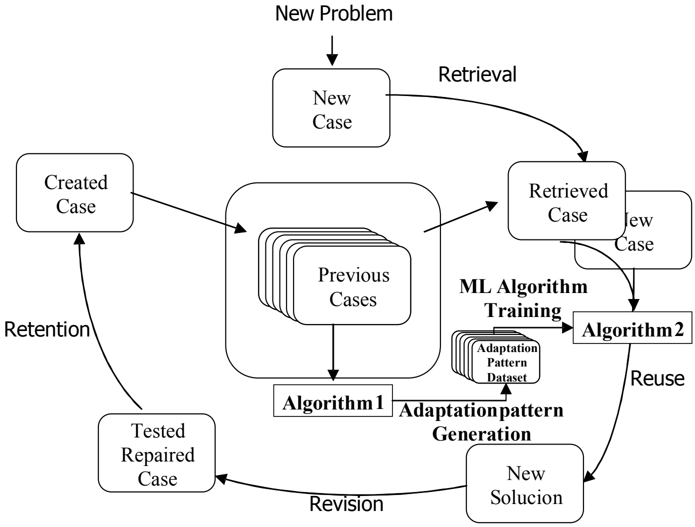


Fig. 2. The CBR CYCLE and the hybrid case adaptation approach (Algorithms 1 and 2).

5 Experimental Results

This Section presents a set of experiments carried out to evaluate the performance of the strategy investigated for automatic case adaptation. For such, the performances obtained with the use of committees of ML algorithms are compared to those obtained by using individual ML algorithms for case adaptation: a MLP nework, a M5 algorithm and a SVM technique. In order to show that the automatic case adaptation may result in considerable gain in the prediction of the desired values for the solution attribute, both case adaptation approaches, using committees of ML algorithms and individual ML algorithms, have their performance compared with the performances obtained by the individual ML algorithms for the prediction of the solution attribute values.

The dataset used in this experiments was employed in the 1999 Computational Intelligence and Learning (COIL) competition, available from the UCI KDD repository [2]. This dataset comes from a water quality study where samples were taken from sites on different European rivers over a period of approximately one year. These samples were analyzed for various chemical substances including: nitrogen in the form of nitrates, nitrites and ammonia, phosphate, pH, oxygen and chloride. In parallel, algae samples were collected to determine the algae population distributions.

There is a total of 340 cases. However, some cases contain missing attributes. After removing cases with missing attributes, 282 cases were felt in the dataset.

The first 11 values of each case are the season, the river size, the fluid velocity and 8 chemical concentrations, which should be relevant for the algae population distribution. The last 7 values of each case are the distribution of different kinds of algae (see Table 1).

Table 1. Water quality case structure.

	Attribute	Values
Problem	Season	summer, spring, autumn, winter
	Size	small, medium, large
	Velocity	high, medium, low
	$CC_{1,1}$	continuous
	…	
	$CC_{1,8}$	continuous
Solution	$AG_{1,1}$	continuous
	…	
	$AG_{1,7}$	continuous

The topology of the MLP networks employed as estimator has 37 input units, a hidden layer with 20 neurons and 1 output neuron. The MLP networks were trained using the momentum backpropagation algorithm, with moment term equal to 0.2 and learning rate equal to 0.3. The M5 algorithm was trained using default parameters. The SVM Algorithm was trained using the Radial Basis Function kernel and default parameters. The MLP networks and M5 algorithm were simulated using the WEKA library, version 3.2 - a set of algorithms of machine learning[2]. The SVM algorithm was simulated using the LIBSVM tool[3]

For the experiments carried out, three different strategies for the generation of adaptation patterns were investigated: strategy I - only the most similar case is retrieved for a given proof case ; strategy II - the three most similar cases are retrieved for a given proof case; strategy III - the five most similar cases are retrieved for a given proof case.

[2] Available in http://www.cs.waikato.ac.nz/ml/weka/index.htm

[3] Available in http://www.csie.ntu.edu.tw/\~cjlin/libsvm

The data normalization (see *Normalization* in Algorithm 2) is only performed if the retrieval and adaptation mechanisms requires it. The cases were stored in the CB in their original format. The numerical values were normalized for the interval [0…1]. For the MLP, SVM and M5 techniques, the input symbolic values were transformed into orthogonal vectors of binary values.

The tests followed the *10-fold-cross-validation* strategy. The patterns were randomly divided into 10 groups (*folds*) with similar size. One *fold* was used as a *test-fold* (a set of new problems to be presented to the system) and the remaining 9 *folds* were considered as a *training-fold* (a set of previously stored cases). Before the training of a ML algorithm using the *training-fold*, the over-sampling technique was applied to the *training-fold.* After the training of a ML algorithm using the over-sampled *training-fold*, the *test-fold* was presented to the system and the average absolute error was calculated. This process was repeated for the remaining 9 *folds.* Next, the average and standard deviation of the absolute error for each training session were calculated. In order to confirm the performance of the hybrid approach, the authors used the *t* test for bilateral procedures which *99%* of certainty [13].

Table 2 shows the results of the tests carried out with the hybrid CBR systems, with individual classifiers and with committees, using the three strategies of adaptation pattern generation. The results obtained by the individual techniques employed alone (MLP, M5 and SVM) are also shown. The best results and the best model are shown in bold.

Table 2. Average error results for the proposed approach. The model **CBR (CSVM - II),** for example, means a CBR system using a **C**ommittee with a **SVM** algorithm as a combiner and using an adaptation pattern data set generated employing the strategy **II** for pattern generation.

Model	Average Absolute Error						
	Algal 1	Algal 2	Algal 3	Algal 4	Algal 5	Algal 6	Algal 7
CBR (M5 – I)	9,86±1,71	6,51±0,65	4,34±0,45	1,38±0,17	5,24±0,60	6,66±0,86	2,38±0,50
CBR (M5 – II)	9,56±2,20	7,13±0,91	4,40±0,63	1,57±0,25	5,24±0,60	6,90±1,45	2,62±0,39
CBR (M5 – III)	10,42±2,54	8,54±1,74	4,64±0,74	1,71±0,25	6,29±1,48	7,14±1,42	2,64±0,54
CBR (SVM – I)	11,42±0,85	7,55±0,57	4,61±0,43	1,44±0,12	5,58±0,33	7,22±0,72	3,19±0,25
CBR (SVM – II)	11,30±0,97	7,40±0,62	4,57±0,51	1,38±0,14	5,61±0,36	6,75±0,80	3,14±0,25
CBR (SVM – III)	11,38±1,08	7,38±0,64	4,56±0,53	1,37±0,15	5,62±0,40	6,79±0,89	3,12±0,26
CBR (MLP – I)	27,19±3,29	8,96±1,33	10,66±1,24	3,25±0,53	11,61±1,80	11,76±1,11	5,86±0,90
CBR (MLP – II)	24,30±5,39	7,86±0,88	8,46±0,81	3,04±0,50	9,35±1,03	1,23±1,64	5,67±0,83
CBR (MLP – III)	26,49±4,16	13,92±1,35	8,94±2,16	3,08±0,52	10,25±1,53	12,02±1,62	5,77±0,88
CBR (CM5 – I)	14,14±1,91	12,03±2,19	7,69±1,29	2,19±0,49	9,48±1,53	9,15±0,75	2,38±0,50
CBR (CM5 – II)	14,56±2,92	11,00±1,46	5,64±0,50	2,02±0,36	7,23±1,01	10,01±1,26	2,62±0,39
CBR (CM5 – III)	15,98±2,43	10,72±1,58	6,30±1,42	2,13±0,42	7,75±1,21	9,20±1,16	2,64±0,54
CBR (CSVM – I)	5,05±0,59	**3,49±0,45**	1,86±0,23	0,63±0,07	**2,84±0,38**	4,33±0,24	**1,50±0,25**
CBR (CSVM – II)	**3,46±0,67**	3,77±0,38	**1,47±0,26**	**0,53±0,10**	3,39±0,30	**3,80±0,33**	1,77±0,17
CBR (CSVM – III)	3,34±0,53	3,92±0,18	1,93±0,20	0,71±0,08	3,48±0,24	3,80±0,40	1,56±0,22
CBR (CMLP– I)	13,99±2,25	11,82±2,04	7,45±1,18	2,19±0,41	9,59±1,41	10,08±0,91	5,02±0,73
CBR (CMLP– II)	16,19±4,22	12,07±1,34	6,24±0,70	2,40±0,42	8,07±0,91	10,82±1,22	4,23±0,45
CBR (CMLP– III)	16,44±2,11	12,29±1,34	6,74±0,98	2,53±0,48	8,07±0,92	10,68±1,59	4,69±0,77
M5	11,23±0,80	6,34±0,59	4,43±0,40	1,38±0,18	5,03±0,57	6,55±0,87	2,34±,48
SVM	10,14±1,26	7,39±0,59	4,62±0,44	1,40±0,13	5,52±0,41	6,86±0,75	3,17±0,24
MLP	23,05±2,70	9,43±1,47	9,98±1,10	3,20±0,62	12,41±2,52	14,06±2,52	5,86±1,92

In order to show the significance of the obtained results, Table 3 exhibits the solution domain values of the dataset.

Table 3. Domain values for the water quality dataset.

Description	Value of the Solution Domain						
	Algal 1	Algal 2	Algal 3	Algal 4	Algal5	Algal6	Algal7
Maximum Value	89,80	72,60	42,80	14,30	61,10	70,0	31,60
Minimum Value	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Average Value	15,13±19,71	7,58±11,03	4,45±8,81	1,36±2,22	6,12±8,74	6,84±12,13	2,24±5,05

The results show that the hybrid approaches, in general, were better in the prediction of the problems solution than the classifiers techniques used alone. The results also show the potential of the hybrid approaches combining inductive learning and instance based learning and suggest that the adaptation pattern data set extracted from the CB contains a good representative sample of the required adaptations over the solution components in the solution space. Additionally, the results show that the performance of CBR systems increases according to the number of retrieved cases in each strategy of pattern generation (strategies I, II and III respectively). This result occurs possibly due to the fact that, in general, a large number of cases produce a high number of adaptation rules. Moreover, the results show that CBR employing committee of ML algorithms introduces more accuracy and stability to the system, by reducing the average absolute error and the standard deviation.

6 Conclusions

This work investigated the use of a hybrid Case Based Reasoning System Approach for monitoring water quality using chemical indicators and algae population. Preliminary results show that the use of a hybrid committee introduces more stability to the system, reducing the standard deviation of the results. The case adaptation approach can be easily extended for domains where the solution of the cases has more than one component, by creating one independent adaptation dataset and one independent ML algorithm or committee for each component of the solution, handling these components as distinct problems. The investigated approach employs a process of adaptation pattern generation that can reduce the effort for knowledge acquisition. Besides, the hybrid approach is not computationally expensive, since the generation of the adaptation patterns demands no comparisons between solution components. Moreover, the process to obtain an adaptation pattern dataset is fully integrated with the case retrieval mechanism and can be implemented employing usual retrieval approaches. The results obtained suggest that the set of adaptation rules extracted from the CB used is consistent and this new approach of adaptation knowledge learning may be a promising technique to solve others real-world problems.

Acknowledgments. The authors would like to thank CNPq, CAPES and FAPESP, Brazilian Research Agencies, for the support received.

References

1. Aamodt,A., Plaza,E.: Case Based Reasoning: Foundational issues, methodological variations, and systems approaches. AI Communications. **7** (1994) 39—59

2. Blake,C.L. and Merz,C.J.: UCI Repository of machine learning databases. http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences (1998)
3. Corchado,J., Lees,B., Fyle,C., Ress,N., Aiken,J.: Neuro-adaptation method for a case-based reasoning system. Computing and Information Systems Journal. **5** (1998) 15—20
4. Devogelaere,D., Rijckaert,M: Third International Competition: Protecting rivers and streams by monitoring chemical concentrations and algae communities solved with the use of GAdC. www.erudit.de/erudit/events/fc-ttc/ erudit-ttc-1-13063.PDF (1999)
5. Duda,R., hart,P., Stork,D.: Pattern Classification. Wiley-Interscience. (2001)
6. Haykin,S.: Neural Networks: A Comprehensive Foundation. Prentice Hall. (1999)
7. Hilario,M.: An Overview Of Strategies For Neurosymbolic Integration. Connectionist-Symbolic Integration: From Unified to Hybrid Approaches. Chapter 2. Lawrence Earlbaum Associates, Inc. (1997)
8. Kolodner,J.: An introduction to case based reasoning. AI Review. **6** (1992) 3—34
9. Leake,D.: CBR in context: The present and future. Case-Based Reasoning: Experiences, Lessons and Future Directions. Chapter 1. AAAI Press/MIT Press. (1996) 1—35
10. Lenz,M., Burkhard,H.-D.: Case Retrieval Nets: Basic Ideas and Extensions. 4th. German Workshop on Case-Based Reasoning: System Development and Evaluation, Berlin, German. Burkhard,H.-D. and Lenz,M. eds. (1996) 103—110
11. Malek,M.: A connectionist indexing aproach for CBR systems. 1st International Conference on Case-Based Reasoning. Sesimbra, Portugal. Veloso,M., Aamodt,A. eds. Springer Verlag (1995) 520—527
12. Malek,M.: Hybrid approaches for integrating neurai networks and case-based reasoning: From loosely coupled to tightly coupled models. Soft Computing in Case Based Reasoning. Chapter 4. Springer Verlag. (2001)
13. Mason,R., Gunst,R., Hess,J.: Statistical design and analysis of experiments. John Wiley & Sons. (1989)
14. Orr,M.: Introduction to radial basis function networks, Technical report. Centre for Cognitive Science. University of Edinburgh. (1996).
15. Policastro,C., Carvalho,A. Delbem,A. : Hybrid Approaches for Case Retrieval and Adaptation. 26th German Conference on Artificial Inteligence. Hamburg, German. Günter,A. et al eds. Springer Verlag (2003) 520—527
16. Policastro,C., Carvalho,A. Delbem,A. : Hybrid Approaches for Case Adaptation. To be published in the proceedings of the 3rd International Conference on Hybrid Intelligent Systems. Melbourne, Australia. (2003)
17. Quinlan,R.: Learning with Continuous Classes. 5th. Australian Joint Conference on Artificial Inteligence, Hobart, Tasmania, World Scientific, Singapore. (1992) 343—348
18. Reategui,E., Campbell,J.: A classification system for credit card transaction. 2th European Workshop on Case-Based Reasoning. Chantilly, France. Keane,M. ed. Springer Verlag. (1994) 280—291
19. Smyth,B.: Case Base Maintenance. 12th. International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Cairo, Egypt. Mira,J., Pobil,A. eds. Springer Verlag (1998) 507--516
20. Vapnik,V.: Statistical Learning Theory. John Wiley & Sons. (1998)
21. Wang,Y., Witten,I.: Induction of model trees for predicting continuous classes. 9th European Conference on Machine Learning. Prague, Czech Republic. Someren,M., Widmer,G. eds. Springer Verlag. (1997) 128—137
22. Watson,I.: CBR is a methodology not a technology. Knowledge-Based Systems.**12** (1999) 303—308

Constructive Meta-learning with Machine Learning Method Repositories

Hidenao Abe[1] and Takahira Yamaguchi[2]

[1] Graduate School of Science and Technology, Shizuoka University
hidenao@ks.cs.inf.shizuoka.ac.jp,
[2] Faculty of Information, Shizuoka University
3-5-1 Johoku Hamamatsu Shizuoka, 432-8011, JAPAN
yamaguti@cs.inf.shizuoka.ac.jp

Abstract. Here is discussed what is constructive meta-learning and how it goes well compared with selective meta-learning that already becomes popular. Selective meta-learning takes multiple learning schemes with the following different ways: bagging, boosting, cascading and stacking methods. On the other hand, constructive meta-learning constructs the learning scheme proper to a given data set. We have implemented constructive meta-learning by recomposing methods into learning schemes with mining (inductive learning) method repositories that come from decomposition of popular mining algorithms. To evaluate our constructive meta-learning, we have done the comparison of the performances of our constructive meta-learning and those of two stacking methods, using UCI/ML common data sets. It has shown us that our constructive meta-learning goes better than the two stacking methods. Furthermore, it turns out to be promising that we apply constructive meta-learning to meta-learner in selective meta-learning.

1 Introduction

In recent years, there are many studies about meta-learning scheme, which enhances performances of classification tasks such as their accuracy, precision, recall and robustness. Current meta-learning scheme contains two different approaches to predict given test data set as shown in Fig.1.

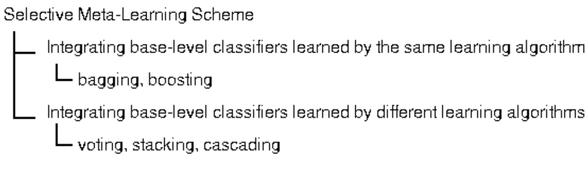

Selective Meta-Learning Scheme
Integrating base-level classifiers learned by the same learning algorithm
bagging, boosting
Integrating base-level classifiers learned by different learning algorithms
voting, stacking, cascading

Fig. 1. Classification of selective meta-learning algorithms

One integrates different base-level classifiers learned by different training data sets with meta-model. This approach contains two well-known methods: boosting[10] and bagging[2]. The other integrates different base-level classifiers learned by different base-level learning algorithms with meta-model or meta-level classifiers. This

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 502-511, 2004.

approach contains three representative methods: voting, cascading[12] and stacking[24].

Although these meta-learning algorithms work well to most data sets, they can not work well to a given data set, when any base-level learning algorithms can not learn a good model to the data set. This problem is caused by the reason why they do not decompose base-level learning algorithms. Since these meta-learning algorithms just select base-level classifiers, we call them “selective meta-learning scheme”.

To above problem, we propose another meta-learning scheme called “constructive meta-learning scheme”. With this approach, we decompose base-level algorithms, and recompose an adequate learning algorithm to a given data set. To implement this scheme, firstly, we have organized an inductive method repository, analyzing representative inductive learning algorithms. Then we have implemented a tool for doing constructive meta-leaning based on the method repository called CAMLET. CAMLET searches for an adequate inductive learning algorithm called an ‘inductive application’, recomposing possible inductive learning algorithms by referring to the method repository.

After implementing CAMLET, we have done an experiment to evaluate the performance of CAMLET as meta-learning scheme on 34 UCI/ML benchmark data sets[4], comparing with two stacking methods implemented in Weka[23].

2 Constructive Meta-learning

The basic idea of constructive meta-learning consists of decomposing base-level algorithms and recomposing an adequate learning algorithm to a given data set. Constructive meta-learning scheme composes an adequate learning algorithm to given data set and user’s requirement as shown in Fig. 2.

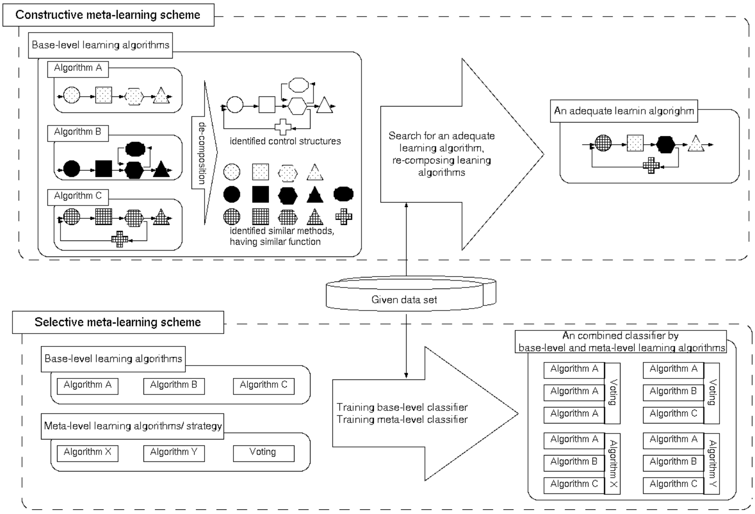

Constructive meta-learning scheme
Base-level learning algorithms
Algorithm A
Algorithm B
Algorithm C
de-composition
identified control structures
identified similar methods, having similar function
Search for an adequate learning algorithm, re-composing leaning algorithms
An adequate learnin algorighm
Given data set
Selective meta-learning scheme
Base-level learning algorithms
Algorithm A
Algorithm B
Algorithm C
Meta-level learning algorithms/ strategy
Algorithm X
Algorithm Y
Voting
Training base-level classifier
Training meta-level classifier
An combined classifier by base-level and meta-level learning algorithms
Algorithm A
Algorithm A
Algorithm A
Voting
Algorithm A
Algorithm B
Algorithm C
Voting
Algorithm A
Algorithm B
Algorithm C
Algorithm X
Algorithm A
Algorithm B
Algorithm C
Algorithm Y

Fig. 2. Overview of selective meta-learning scheme and constructive meta-learning scheme.

On the other hand, selective meta-learning scheme just combines base-level classifiers. They learn base-level classifiers by training data sets. Then some algorithms combine them with simple voting or weighted voting. Others learn a meta-level classifier for classifying better base-level classifier or the class of test instances. The meta-level classifier is learned by meta-level training data set made from the predictions of base-level classifiers to validation data sets, which are generated from the training data set.

Constructive meta-learning scheme has the following three major issues:

1. how to decompose base-level learning algorithms into parts
2. how to restrict combinations of these parts because they work better as learning algorithms than base-level learning algorithms
3. how to recompose an adequate learning algorithms to given data set

We have solved these issues with constructing a method repository and searching the possible learning algorithms for an adequate one. So we call this approach "constructive meta-learning based on method repository".

3 An Implementation of Constructive Meta-learning Based on Method Repository

To implement constructive meta-learning based on method repository, we have constructed a method repository, analyzing inductive learning algorithms. The method repository includes an organization of each part of algorithms, control structure templates to reconstruct learning algorithms and a data type hierarchy to organize the methods. With this repository, CAMLET automatically recomposes inductive applications, and searches for an adequate inductive application to a given data set.

3.1 Constructing an Inductive Learning Method Repository

To construct the method repository, firstly, we have analyzed representative inductive learning algorithms. Then we have identified similar functional part called inductive learning methods. Each group consists of one or more similar specific methods, and is called generic method. The control structure templates have been defined to these generic methods.

We have analyzed the following representative learning algorithms: Version Space[18], AQ15[17], ID3[20], C4.5[21], Classifier Systems[5], Back Propagation Neural Network[14], Bagged C4.5 and Boosted C4.5[22]. The analysis result first came up with just unstructured documents to articulate what inductive learning methods are in the above eight learning algorithms. Although it is a hard issue to decide a proper grain size of methods, we analyzed under the condition that inputs and outputs are data sets or/and classifier sets. Thus we have identified 25 specific methods from the eight representative inductive learning algorithms.

By grouping similar methods, we have identified the following six generic methods: "generating training and validation data sets", "generating a classifier set", "evaluating a classifier set", "modifying training and validation data sets", "modifying a classifier set" and "evaluating classifier sets to test data set". To these six generic methods, we have defined eight control structures with three feed back loops to reconstruct inductive learning algorithms, as shown in Fig. 3.

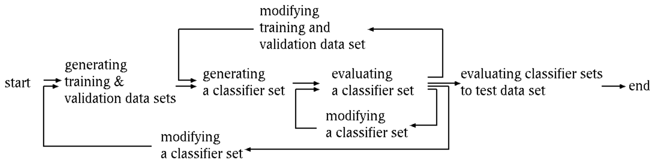

start
generating training & validation data sets
modifying training and validation data set
generating a classifier set
evaluating a classifier set
evaluating classifier sets to test data set
end
modifying a classifier set
modifying a classifier set

Fig. 3. The six generic methods and eight control structures with three feed back loops.

We have organized the generic methods and the specific methods into a hierarchy as shown in Fig.4, characterizing each generic method with input data type, output data type, reference data type, pre-method and post-method. In Fig.4, the six generic methods are put in the left column named 'L1', and each leaf node represents each specific method identified from representative learning algorithms. Thus leaf nodes of the hierarchy get into each specific method from some learning algorithm and its executable code written in C language. For example, the method of "a void validation data set" does not split given training data set into internal training and validation data sets. So the validation data set becomes given training data set.

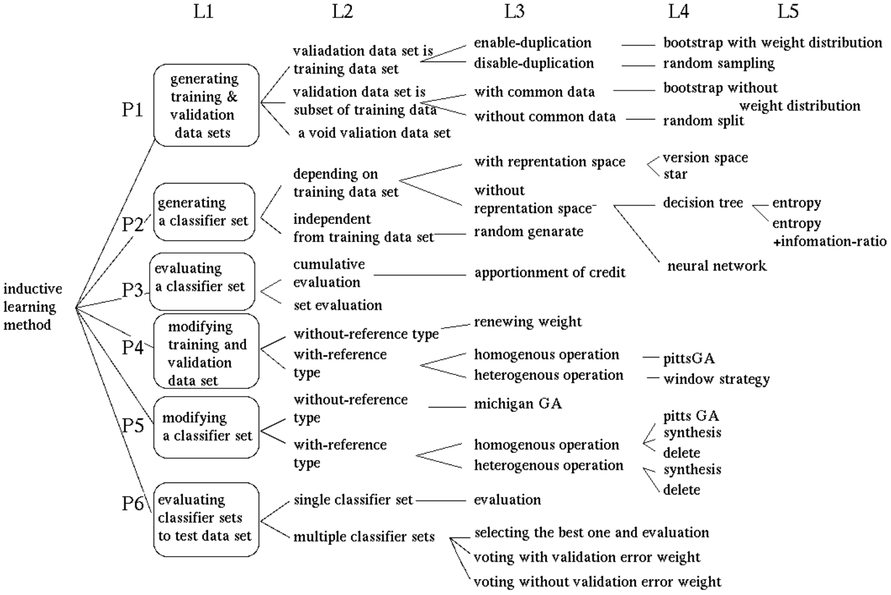

L1
L2
L3
L4
L5
inductive learning method
P1
generating training & validation data sets
valiadation data set is training data set
enable-duplication
bootstrap with weight distribution
disable-duplication
random sampling
validation data set is subset of training data
with common data
bootstrap without weight distribution
without common data
random split
a void valiation data set
P2
generating a classifier set
depending on training data set
with reprentation space
version space
star
without reprentation space
decision tree
entropy
entropy +infomation-ratio
neural network
independent from training data set
random genarate
P3
evaluating a classifier set
cumulative evaluation
apportionment of credit
set evaluation
P4
modifying training and validation data set
without-reference type
renewing weight
with-reference type
homogenous operation
pittsGA
heterogenous operation
window strategy
P5
modifying a classifier set
without-reference type
michigan GA
with-reference type
homogenous operation
pitts GA
synthesis
delete
heterogenous operation
synthesis
delete
P6
evaluating classifier sets to test data set
single classifier set
evaluation
multiple classifier sets
selecting the best one and evaluation
voting with validation error weight
voting without validation error weight

Fig. 4. Hierarchy of the method repository

To support making the hierarchy of the method repository, we have also defined a data type hierarchy as shown in Fig.5. These data types are manipulated by the methods.

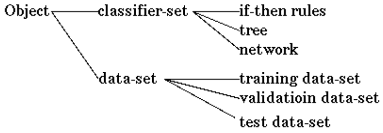
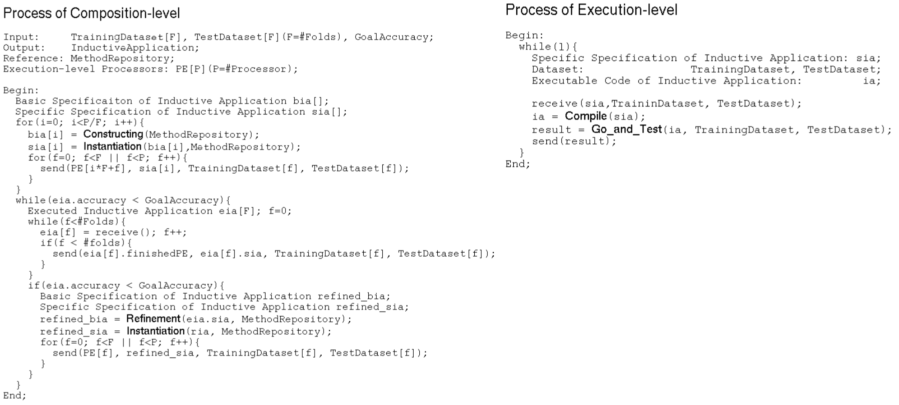


Fig. 5. Data type hierarchy

3.2 A Tool for Doing Constructive Meta-learning Based on Method Repository

We have designed parallel CAMLET as shown in Fig.6. The inputs of CAMLET are training data set, test data set, and goal accuracy. The output of CAMLET is an inductive application proper to the given data sets. CAMLET searches for an adequate inductive application, executing inductive applications yielded by the method repository. We have parallelized executions of inductive applications, because these executions have high computational cost. Parallel CAMLET takes one processing element named ‘Composition-level’ to compose specifications of inductive applications and to control, and other processing elements named ‘Execution-level’ to execute inductive applications.



Process of Composition-level

```
Input:     TrainingDataset[F], TestDataset[F](F=#Folds), GoalAccuracy;
Output:    InductiveApplication;
Reference: MethodRepository;
Execution-level Processors: PE[P](P=#Processor);

Begin:
  Basic Specificaiton of Inductive Application bia[];
  Specific Specification of Inductive Application sia[];
  for(i=0; i<P/F; i++){
    bia[i] = Constructing(MethodRepository);
    sia[i] = Instantiation(bia[i],MethodRepository);
    for(f=0; f<F || f<P; f++){
      send(PE[i*F+f], sia[i], TrainingDataset[f], TestDataset[f]);
    }
  }
  while(eia.accuracy < GoalAccuracy){
    Executed Inductive Application eia[F]; f=0;
    while(f<#Folds){
      eia[f] = receive(); f++;
      if(f < #folds){
        send(eia[f].finishedPE, eia[f].sia, TrainingDataset[f], TestDataset[f]);
      }
    }
    if(eia.accuracy < GoalAccuracy){
      Basic Specification of Inductive Application refined_bia;
      Specific Specification of Inductive Application refined_sia;
      refined_bia = Refinement(eia.sia, MethodRepository);
      refined_sia = Instantiation(ria, MethodRepository);
      for(f=0; f<F || f<P; f++){
        send(PE[f], refined_sia, TrainingDataset[f], TestDataset[f]);
      }
    }
  }
End;
```

Process of Execution-level

```
Begin:
  while(1){
    Specific Specification of Inductive Application: sia;
    Dataset:                     TrainingDataset, TestDataset;
    Executable Code of Inductive Application:         ia;

    receive(sia,TraininDataset, TestDataset);
    ia = Compile(sia);
    result = Go_and_Test(ia, TrainingDataset, TestDataset);
    send(result);
  }
End;
```

Fig. 6. Pseudo code for the algorithm of parallel CAMLET

At the *construction* activity, CAMLET constructs basic specifications of inductive applications randomly by selecting a path from “start” to “end” in Fig.3 and specific methods in Fig.4. At the *instantiation* activity, CAMLET instantiates a basic specification, checking connections of specific methods with roles of each method. When a method is connected to another method at the level of specific method, the specification for I/O data types must be unified. To do so, the instantiation activity inserts a data type conversion procedure such as converting a decision tree to its rule set. Then CAMLET sends instantiated specifications to execution-level processing elements. At the *compilation* activity, CAMLET compiles received specific specification into executable code by referring to the method repository. The *go and test* activity executes actual inductive application to given data set, and tests with given test data set. Then the execution result is send to composition-level processor. If the result accuracy can not go to or beyond input goal accuracy, CAMLET *refine*s the specification with GA operations as shown in [1].

4 Experiments on UCI/ML Common Data Sets

After implementing the method repository, which has been constructed from eight representative inductive learning algorisms, and parallel CAMLET written in Perl and C language, we have done the experiment to answer the following two questions:

- Does CAMLET have good performance as a meta-learning algorithm?
- Does CAMLET enhance the performance of classification tasks, comparing with its base-level learning algorithms?

To the first question, we have compared the performance of inductive applications composed by CAMLET with that of two stacking methods of Weka 3.3.6 on 34 data sets from UCI/ML repository. To the second question, we also compared the performance from above result with that of base-level classifiers to each method.

4.1 Datasets and Evaluation Results

The evaluation method for UCI/ML data sets is 10-fold cross validation. Each pair of training and test data set have been generated by SplitDatasetFilter implemented in Weka 3.3.6, setting up the same random seed (-S =1) to each data set.

As the base-level learning algorithms of two stacking methods, we have set up the following six algorithms: J4.8, Part rule set learner[9], IBk, Naïve Bayes, Bagged J4.8 with 10 iterations, Boosted J4.8 up to 10 iterations. As the meta-level learning algorithms of each stacking method, we have set up J4.8 and 'Classification via Linear Regression', which we call CLR later.

Goal accuracies of each data set for CAMLET have been set up the highest accuracy of the two stacking methods. CAMLET has searched for an adequate inductive application until going to or beyond the goal accuracy, executing them to each data set. If any adequate inductive application has not been able to find out after one hundred executions, CAMLET has selected just one inductive application, which has the highest accuracy and the least computational cost within them.

Table 1 shows the average classification accuracy for every method and data set, and indicates significant win/loss compared to each inductive application composed by CAMLET. Table 2 shows numbers of significant win/loss of rows' meta-learning methods compared to column base-level algorithms. To test the significance of the difference of the average accuracy, we have applied standard t-test with accuracies of each fold of cross validation on each data set.

Table 1. Average classification accuracy(%) and its standard deviation.

	CAMLET		Stackin(CLR)			Staking(J4.8)		
	Acc.	SD	Acc.	SD	win/loss	Acc.	SD	win/loss
anneal	99.56	0.78	99.44	0.59		99.55	0.78	
audiology	73.97	12.33	69.60	16.60		72.15	12.04	
autos	43.86	15.72	39.38	20.78		42.26	18.29	
balance-scale	86.57	3.32	89.45	3.08		88.82	6.62	
breast-cancer	73.82	7.03	72.07	7.87		73.46	7.73	
breast-w	97.14	3.08	96.57	3.88		96.28	4.11	
colic	83.44	7.23	85.05	5.17		84.21	6.44	
credit-a	84.78	18.61	83.48	19.24		82.75	19.77	
credit-g	74.90	3.96	75.00	4.78		73.80	5.43	
diabetes	76.18	6.17	75.52	6.33		73.83	7.22	
glass	70.22	9.77	67.38	19.79		70.56	9.96	
heart-c	84.50	7.91	64.40	20.08	loss	76.91	5.54	loss
heart-h	80.40	19.14	46.09	40.16	loss	78.09	14.92	
heart-statlog	84.45	6.00	83.70	5.58		84.07	4.29	
hepatitis	83.63	11.57	83.08	10.17		76.38	14.38	
hypothyroid	99.58	0.18	99.50	0.26		99.58	0.26	
ionosphere	93.45	4.26	93.17	7.00		91.77	7.08	
iris	95.33	5.49	92.67	11.09		92.00	8.78	
kr-vs-kp	97.87	2.56	97.75	2.33		96.56	4.22	
labor	89.33	17.27	91.67	8.78		89.67	8.95	
letter	94.59	0.34	96.82	0.31	win	95.35	0.63	win
lymph	81.76	10.45	81.10	8.75		85.10	6.26	
mushroom	100.00	0.00	99.93	0.23		100.00	0.00	
primary-tumor	40.13	7.40	40.11	6.89		38.05	8.11	
segment	97.62	1.24	98.10	1.10		97.79	1.28	
sick	99.12	0.62	99.05	0.50		98.89	0.41	
sonar	61.60	13.16	53.50	22.30		53.50	21.54	
soybean	88.84	9.83	89.28	9.46		86.52	8.46	
splice	89.56	7.91	94.92	1.82		94.11	2.64	
vehicle	76.00	2.82	75.40	3.40		74.69	3.95	
vote	96.07	2.70	96.54	2.51		95.39	3.64	
vowel	65.16	7.13	64.55	5.94		59.29	5.09	
waveform-5000	82.14	1.31	84.30	1.14	win	83.48	1.80	
zoo	96.00	5.16	93.00	6.75		96.00	6.99	
Average	83.58	6.84	81.52	8.37		82.38	6.99	

Table 2. Number of win/loss(win:loss) where algorithm in row significantly outperforms algorithms in column

CAMLET vs. its base-level algorithms

	C4.5 (unpruned)	ID3 (unpruned)	Classifier Systems	Neural Network	Boosted C4.5	Bagged C4.5
CAMLET	12:0	15:0	21:0	19:0	7:0	5:0

Stacking vs. its base-level algorithms

	J4.8	Part	Na□ve Bayes	IBk(k=5)	Boosetd J4.8	Bagged J4.8
CLR	6:0	8:1	12:3	10:2	3:0	5:1
J4.8	7:0	6:0	10:1	9:0	2:0	3:0

4.2 Discussion of Results

To answer the first question, the performance of inductive application composed by CAMLET has been higher than two stacking methods. As shown in Table 1, inductive applications composed by CAMLET take the first place to the following 22 data sets: anneal, audiology, autos, breast-cancer, breast-w, credit-a, diabetes, heart-c, heart-h, heart-statlog, hepatitis, hypothyroid, ionosphere, iris, kr-vs-kp, mushroom, primary-tumor, sick, sonar, vehicle, vowel, zoo. To heart-c, the inductive application composed by CAMLET has significantly outperformed the two stacking methods. Looking at the number of significant win/loss, CAMLET shows as good performance as the best performance of the two stacking methods. Although CAMLET has not been able to recompose any stacking method, CAMLET has achieved higher performance than each stacking method, only combining control structures and methods extracted from base-level learning algorithms.

To the second question, the number of significant win/loss in Table 2 shows us the ability as a meta-learning scheme. As shown in Table 2, CAMLET has never lost any base-level learning algorithm of itself. Additionally, CAMLET has won more data sets than the two stacking methods. As this result, we can say that our “constructive meta-learning based on method repository” has satisfactory ability as a meta-learning method.

5 Related Work

There is another study of selective meta-learning scheme. In [6] and [11], they present dynamic selection of base-level classifiers to a given test data set with meta-rule, which is learned with measurable characteristics of given data set. [16] and other ranking methods[7] are also worked as selective meta-learning scheme. To support data mining algorithm selection, they make the ranking of base-level algorithms to given data set based on heuristic scores.

To support whole data mining process including data pre-processing, data mining, post-processing of the mining results, there is a way to present the ranking of data mining applications to given data set like IDA[3]. However, if users do not get a perfect ranking, they need to execute the data mining applications for clearing on their actual performance.

Many data mining tools are developing to execute data mining algorithms for various data sets. Weka is one of the platforms for constructing adequate data mining applications with a useful interface. However, these tools only supply an execution environment and literal information of each part of data mining process. Weka is also used as a data mining method library like MLC++[15]. Although data mining methods are now standardizing with PMML[19], their methods including Weka and MLC++ are far coarser than that of CAMLET.

As the work for constructing data mining application, MEDIA-model[8] is a reference structure for the application of inductive learning techniques. It focuses on methodological aspects of data mining applications but not on automatic composition facilities of data mining applications like CAMLET.

6 Conclusions

We present an evaluation of our proposal called "constructive meta-learning based on method repository". To construct the method repository, we have analyzed eight representative inductive learning algorithms. With this analysis, we have identified 25 specific inductive learning methods. To organize them to the method repository, we have identified six generic methods and eight kinds of control structures. Then parallel CAMLET has designed to search the specification space of inductive applications yielded by the method repository for an adequate one faster.

After implementing the method repository and parallel CAMLET, we have done an experiment to evaluate our approach with 34 UCI/ML common data sets, comparing with two stacking methods and their base-level learning algorithms. As the result, CAMLET shows us higher performance than the two stacking methods. Comparing with the performance of their each base-level learning algorithm, CAMLET also shows us significant improvements as a meta-learning scheme. With our constructive meta-learning, it is also possible to select proper base-level and meta-level learning algorithms for a selective meta-learning.

As a future work, we will extend CAMLET to support whole data mining process development. To construct data mining applications more flexibly, we will develop the following new repositories: a data pre-processing method repository and a metrics repository, which is to handle multiple criteria[13] with explicit model. In addition, we will extend the inductive learning method repository to the mining method repository. Although we have not search parameters included in mining methods and control structures for the best combination, we need to search them, because they affect the performance of data mining applications. Since the specification space of data mining applications will become explosively larger space with above extensions, we will also develop more efficient search method to search the specification space of data mining applications.

References

1. Abe, H., and Yamaguchi, T.: "Comparing the Parallel Automatic Composition of Inductive Applications with Stacking Methods", in Proc. of the ECML/PKDD-2003 Workshop on Parallel and Distributed Computing for Machine Learning, pp.1—12 (2003).
2. Breiman, L.: "Bagging Predictors", Machine Learning, 24(2), pp.123-140 (1996).
3. Bernstein, A., and Provost, F.: "An Intelligent Assistant for Knowledge Discovery Process", IJICAI 2001 Workshop on Wrappers for Performance Enhancement in KDD, (2001)
4. Blake C. L., and Merz, C. J.: UCI/ML repository of machine learning databases, (1998). [http://www.ics.uci.edu/~mlearn/MLRepository.html]
5. Booker, L. B., Holland, J. H., and Goldberg, D. E.: "Classifier Systems and Generic Algorithms", Artificial Intelligence, 40, pp.235-282 (1989).
6. Brazdil, P., Gama, J., and Henery, B.: "Characterizing the Applicability of Classification Algorithms Using Meta-Level Learning", in Proc. of the European Conference on Machine Learning (ECML-94), pp. 83—102 (1994).
7. Brazdil, P., and Soares, C.: "A Comparison of Ranking Methods for Classification Algorithm Selection", in Proc. 11th European Conference on Machine Learning (ECML-2000), pp. 63—74 (2000).
8. Engels, R.: "Planning in Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance", Institute for AIFB (1996).
9. Frank, E., and Witten, I. H.: "Generating Accurate Rule Sets without Global Optimization", in Proc. of Fifteenth International Conference on Machine Learning, pp. 144--151 (1998).
10. Freund, Y., and Schapire, R. E.: "Experiments with a new boosting algorithm", in Proc. of Thirteenth International Conference on Machine Learning, pp. 148--156 (1996).
11. Gama, J., Brazdil, P.: "Characterization of Classification Algorithms", 7th Portuguese Conference on Artificial Intelligence, EPIA '95, pp. 189--200 (1995).
12. Gama, J. and Brazdil, P.: "Cascade Generalization", Machine Learning 41(3), Kluwer Academic Publishers, Boston, pp.315--343, (2000).
13. Giraud-Carrier, C.: "Beyond predictive accuracy: what?", in Proc. of Upgrading Learning to the Meta-Level: Model Selection and Data transformation, pp. 78--85 (1998).
14. Hinton, G. E.: "Learning distributed representations of concepts", in Proc. of 8th Annual Conference of the Cognitive Science Society, pp.1--12 (1986).
15. Kohavi, R., and Sommerfield, D.: "Data Mining using MLC++ | A Machine Learning Library in C++", in Proc. 8th International Conference on Tools with Artificial Intelligence, pp. 234--245 (1996).
16. Metal Project: [http://www.metal-kdd.org/]
17. Michalski, R., Mozetic, I., Hong, J. and Lavrac, N.: "The AQ15 Inductive Learning System: An Over View and Experiments", Reports of Machine Learning and Inference Laboratory, No.MLI-86-6, George Mason University (1986).
18. Mitchell, T. M.: "Generalization as Search", Artificial Intelligence, 18(2), pp.203-226 (1982).
19. PMML (Predictive Model Markup Language): [http://www.dmg.org/]
20. Quinlan, J. R.: "Induction of Decision Tree", Machine Learning, Vol.1, Morgan Kaufmann, pp.81--106 (1986).
21. Quinlan, J. R.: "Programs for Machine Learning", Morgan Kaufmann (1992).
22. Quinlan, J. R.: "Bagging, Boosting and C4.5", in Proc. of American Association for Artificial Intelligence, pp.725—730 (1996).
23. Witten, I., and Frank, E.: "Data Mining: Practical machine learning tools and techniques with Java implementations", Morgan Kaufmann Publishers (2000). [http://www.cs.waikato.ac.na/~ml]
24. Wolpert, D.: "Stacked Generalization", Neural Network 5(2), pp.241-260 (1992).

An Algorithm for Incremental Mode Induction

Nicola Di Mauro, Floriana Esposito, Stefano Ferilli, and Teresa M.A. Basile

Dipartimento di Informatica, Università di Bari - Italy
{nicodimauro,esposito,ferilli,basile}@di.uniba.it

Abstract. Learning systems have been devised as a way of overcoming the knowledge acquisition bottleneck in the development of knowledge-based systems. They often cast learning to a search problem in a space of candidate solutions. Since such a space can grow exponentially, techniques for pruning it are needed in order to speed up the learning process. One of the biases used by Inductive Logic Programming (ILP) systems for this purpose is mode declaration. This paper presents an algorithm to incrementally learn this type of meta-knowledge from the available observations, without requiring the final user's intervention.

1 Introduction

The knowledge acquisition process is widely known to be a serious bottleneck in the development of intelligent systems, such as expert systems and decision support systems. For this reason, great interest has been devoted in the implementation of learning systems that are able to automatically induce the needed knowledge from observations. If, on the one hand, efficiency is a critical factor for the success of such systems, on the other hand reaching high performance is often strongly dependent on good parameter settings, which in turn require deep technical knowledge and can be hardly done by the final users. This is why it would be highly desirable to set automatically the parameters of the learning systems.

Inductive Logic Programming (ILP) is an established sub-field of Machine Learning aimed at inducing first-order clausal theories from examples. This objective can be cast to a search problem in a hypotheses space, that may lead to serious efficiency problems when the complexity of the learning task and/or of the description language used causes an explosion in the size of such a space. For this reason, many ILP systems are designed to use, if available, various forms of *meta-knowledge* about the hypotheses to be learned in order to make the search more efficient.

In a general ILP scenario, given a description language, unary predicates represent values of the objects properties, while n-ary predicates represent associations among objects, and hence raise the problem of understanding which of their arguments must be supplied as input and which ones they return as a result of their computation (*mode* of the predicate), according to specific use of the predicates in a given context. Such issue is well-known, for instance, in Prolog, where the procedural interpretation re

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 512-522, 2004.

quires predicates' arguments to be used properly. This holds not only for predicates having a unique use, but also for reversible predicates, in which different combinations of input and output arguments are possible. However, in Prolog the possible uses of built-in and library predicates are pre-defined, or known to the programmer in the case of user-defined ones. The situation is different in first-order machine learning, where the available information (examples, observations, background knowledge, etc.) could be provided by sources that are extraneous to the experimenter, and thus the mode of the predicates in the description language could be unknown.

A *mode declaration* [2] is a specification of the predicate modes. It plays a central role in the efficiency improvement of some systems (e.g., Progol [12], GOLEM [10], FOIL [11]), and it is required, in particular, by systems designed to learn recursive definitions (such as TIM [14] and MRI [13]). This paper presents an algorithm that induces *mode declarations* from examples of the concept to be learned. Its main novelty lays in its *incremental behaviour*, useful when the knowledge is not completely available at the beginning of the learning process: in fact, it works without requiring a previous theory for the target concept. It can be embedded in a natural way in ILP systems as a pre-processor.

In Section 2 some significant previous works on Automatic Mode Inference in an ILP problem are presented. Section 3 introduces the basic concepts needed to understand the proposed approach and presents the new algorithm. In Section 4 the main characteristics and the behavior of the algorithm are presented by means of examples. Finally, Section 5 draws some conclusions.

2 Related Work

Formally, a *mode* of an *n*-ary predicate symbol is an *n*-tuple that represents a possible instantiation of arguments of that predicate symbol in terms of some domain. Each element of such a domain corresponds to a degree of instantiation of an argument of the predicate symbol. In this work we set such a domain to {+,–}, where '+' means that the argument must be instantiated, and '–' expresses the fact that the argument is returned after the predicate computation.

Generally speaking, in a logic program there is not the concept of "input" and "output" variable: each variable might in principle be used either as an input or as an output argument, and programs might be executed in either a "forward" or a "backward" direction. Nevertheless, the need of mode declarations may arise in Logic Programming for different reasons. For instance, knowledge about which arguments in a predicate must be specified and which ones are returned after computation can help programmers in verifying program correctness. Moreover, compilers can profitably use it for optimization purposes (e.g., using efficient special-purpose unification routines instead of the general unification algorithm): indeed, it is often the case that, in a particular program, a predicate is executed in one direction only, i.e. it is always called with a particular set of its variables that are bounded (the "input" variables) and the remaining set unbounded (the "output" variables).

Traditionally, the task of supplying this kind of information has been in charge of the programmer. In such a case, however, problems might arise due to the errors made by the programmer in declaring modes that can lead to very strange program behavior, whose cause can be hard to find. A possible solution may be letting the compiler infer the modes, using them either to optimize a program without mode declarations, or to verify the declarations made by the programmer, in a similar fashion to type-checkers used in other languages to verify type declarations. The issue of automatic mode inference for these purposes has been considered in [3], [4], [5], [6] and [7]. All these works deal with a Prolog program (i.e., a set of definite Horn clauses) together with a negative clause (called *query*), where the mode of a predicate in the program indicates how its arguments will be instantiated when that predicate is called. Thus, the modes of a program represent statements about all computations that are possible from it [8].

Another use of mode declarations, as introduced in Section 1, comes from the ILP where the modes are exploited to build systems that are able to explore the space of hypotheses more efficiently. An algorithm to extract modes from data was presented in [1] and implemented in LIME [15]. In LIME the search space is restricted to *determinate clauses*; in the absence of any information, each time a literal is added to a clause the system must assume that unbounded variables will be uniquely bounded. As this process is essentially the same each time it is carried out, considerable improvement in performance can be achieved if mode information is available. LIME extracts such information from the data, this way being able to skip clauses that are not determinate. It considers predicates one at a time and assumes that any possible mode declaration for that predicate has functional dependency. Then this assumption is checked against the data: if it is false (i.e., the data show a counterexample), then the algorithm discards that possible functional dependency. Another system that faced these issues is MOBAL [9].

Example 1. Consider predicates `append(l1,l2,l3)` ("`l3` is the list obtained appending the list `l2` to the list `l1`") and `decomp(l,e,r)` ("`l` is the list with head `e` and tail `r`"). Mode declarations `{append(+,+,-),append(+,-,+),append(-,+,+),decomp(+,-,-),decomp(-,+,+)}` are not consistent with `append`$(X_1,X_2,X_3) \leftarrow$ `decomp`(X_1,X_2,X_3), and hence the search space can be pruned by eliminating such a clause.

3 MILE: Mode-Declarations Incremental Learner from Examples

In this section, after giving some preliminary definitions and a general idea of the proposed technique for inducing mode declarations from a set of examples for some concept (predicate), we present the detailed algorithm.

Examples are ground Horn clauses and the hypotheses space consists of *function free* clauses, i.e. terms in the head of the target clauses and terms in the literals of the body can be either variables or constants. This simplifies the structure of each clause without affecting expressiveness [16]. Each clause consists of a head literal followed by a list of body literals. Each literal in the body of a clause must use at least one term (variable or constant) that has been introduced by a previous literal in the body. Hence, the list of literals in the body can be ordered according to the following definition.

Definition 1. (Layers of a clause) *Given a linked clause $h \leftarrow b_1,b_2,\ldots,b_n$: the set of* 1-layer literals *consists of all b_i's, $1 \leq i \leq n$, such that b_i shares some argument with h. The arguments of the literal h are called* 1-layer activated terms*; the set of* k-layer literals *consists of all b_i's such that b_i is not a t-layer literal for $1 \leq t < k$ and it shares some of its arguments with the set of (k–1)-layer activated terms. The set* of k-layer activated terms *(also known as* variable depth *[12]) is the union of all k-layer literals' terms that are not (k–1)-layer activated terms. This notion induces an equivalence relation on the set of literals in a given clause.*

Example 2. The literals in the body of the following Horn clause:

```
h(X) :- p(X,Y),p(X,Z),p(X,W),o(Y,Z),q(Z,T),s(T),l(W).
```

can be ordered in layers (equivalence classes). The set of *1-layer activated terms* is `{X}`, that contains the only argument that is present in the head of the clause; `p(X,Y)`, `p(X,Z)`, `p(X,W)` are the *1-layer literals*, since they contain the variable `X` as argument. Hence, the set of *2-layer activated terms* is `{Y,Z,W}`. Then, `o(Y,Z)`, `q(Z,T)`, `l(W)` are the *2-layer literals* and `{T}` is the set of *3-layer activated terms*. Finally, `s(T)` is the only *3-layer literal.*

According to the identified layers, the above clause can be rewritten as follows:

```
h(X) :- [  [ p(X,Y), p(X,Z), p(X,W) ],      % X
           [ o(Y,Z), q(Z,T), l(W) ],        % Y Z W
           [ s(T) ] ].                      % T
```

After introducing how to represent a clause by means of layers, we can give the general idea behind MILE. Such an algorithm can be used as a preprocessor of the examples given as input to an ILP system. It is incremental but obviously it can be exploited also by batch systems.

Given a Horn clause:

1. select a possible mode declaration[1] for the literal in its head and consider the corresponding input arguments as *1-layer activated terms*;
2. for $k \geq 1$ repeat the following:
   1.1. collect the set of *k-layer literals*
   1.2. consider as input arguments those that correspond to *k-layer activated terms* and as output arguments the remaining ones.

It is important to note that, at any step (corresponding to a layer), if an output term is introduced by more than one literal (i.e., there are many *k*-layer literals with the

[1] Given a literal $p(x_1,x_2, \ldots ,x_n)$, there are 2^n-1 possible mode declarations, excluding the mode that considers all arguments as output ones (we assume that there must be always at least one input argument according to which computing the others). For instance, predicate *p/3* has $(2^3$-$1)=7$ possible modes, specifically: (+,+,+),(+,+,–),(+,–,+),(–,+,+),(+,–,–),(–,+,–),(–,–,+).

Algorithm 1. *MD(Bound,Lits,Modes):NewModes*

```
/* Bound: set of terms; Lits: list of literals;
Modes: set of set of mode declarations: Modes = {$m_1$, ...,$m_n$} where $m_i$ is a set of modes*/

if Lits ≠ []
  Bound' ← ∅; Lits' ← Lits
  forall $b_i \in Lits$ that shares some of its arguments with Bound do
    if $b_i$ is not a linked-layer literal then
      $M_i$ is the mode of $b_i$ obtained by setting its arguments that are in Bound as
        input (+) and the others as output (–)
      if consistent($M_i$,Modes) then
        Remove $b_i$ from Lits' and add to Bound' the output terms of $b_i$
      else remove inconsistency from Modes
    else Link ← list of all literals linked-layer to $b_i$
      forall $b_j \in$ Link do
        $M_j$ is the mode of $b_j$ obtained by setting its arguments that are k-layer
          activated terms as input (+) and the others as output (–)
        if consistent($M_i$,Modes) then
          Remove $b_i$ from Lits' and add to Bound' the output terms of $b_j$
        else remove inconsistency from Modes and exit
        if Modes ≠ ∅
          NewModes ← MD(Bound',Lits',Modes)
        else fail
else NewModes ← Modes
return(NewModes)
```

same term marked as "output" by step 1.2), it is not possible to decide which of them actually introduces it. For example, if `p(X,Y)` and `q(X,Y)` are two *k*-layer literals and `X` is a *k*-layer activated term, then it is necessary to assess if `Y` is produced by `p` or by `q`. In these cases, the algorithm maintains all the possible alternatives, waiting for the next steps and/or examples to (hopefully) *disambiguate* the correct one.

Example 3. Given the following clause:

```
   h(a,b) :- p(a,b), p(b,c), o(a,b,c).
```

and hypothesizing mode (+,–) for the head literal, there are two *1*-layer literals, `p(a,b)` and `o(a,b,c)`, that have the same term `b` as a possible output. In this case, the available information was not sufficient to identify a single mode for predicates `p/2` and `o/3` given the mode (+,–) for the predicate `h/2`. Indeed, the list of modes induced by MILE for the above clause is

```
   md([[ h(-,+), [ [p(+,+),o(-,+,-)]]],
        [ h(+,-), [ [p(+,+),o(+,-,-)], [p(+,-),o(+,+,+)]]],
        [ h(+,+), [ [p(+,+),o(+,+,-)]]]]).
```

Algorithm 2. MILE(E)

```
$\boldsymbol{M} = \varnothing$
forall $e \in E$ do
  $(h/n)$ ← predicate in the head of $e$
  forall possible modes $M_{h/n}$ of the predicate $h/n$ do
    Bound ← input terms of $h$
    Lits ← the body of $e$
    if $m(M_{h/n}, MD(Bound, Lits, \varnothing))$ is consistent with $\boldsymbol{M}$
      update $\boldsymbol{M}$ accordingly
    else
      fail
```

Note that a mode is related to the concept described by the predicate it belongs to (i.e., to its context[2]). The term *disambiguation* refers to the assumption that there is a unique mode for each predicate in a given context (i.e., each predicate is used always with the same meaning – it is a function). Thus, the algorithm is able to learn *many modes for the same predicate*, even if just one for each fixed context.

Definition 2. ***(linked-layer literal)*** *A k-layer literal p is called* linked-layer literal *if there exists a k-layer literal q that shares some argument with p and such arguments are not k-layer activated terms; in this case, we also say that p is* linked-layer to *q*.

Algorithm 1 describes how MILE learns mode declarations for the predicates in the body of an example '*h :- Body*'. Let us consider the list $Lits=(l_1,\dots,l_n)$ of such body literals, the set *Bound* of 1-layer activated terms (input arguments of *h*), and an initial set of mode declarations *Modes* (=$\varnothing$ at beginning for a new example). For each literal $l_i \in Lits$ that shares some arguments with the set *Bound*: **a)** if l_i is not a linked-layer literal then its mode considers as input terms all those that are in *Bound*, and as output the others; **b)** if l_i is a linked-layer literal, then all the literals that are linked-layer to it are collected, and all possible modes for them are considered (see Example 3). Each new mode is added to the set *Modes*, and all elements of *Modes* that are inconsistent[3] with it are eliminated.

If many examples are available, they can be used to further refine the identified set of modes, as described in Algorithm 2. At any moment, MILE maintains a set $\boldsymbol{M}$ of all consistent modes learned thus far for all concepts and predicates encountered in the processed examples. For each new incoming example, MILE applies Algorithm 1 to find all possible modes for its predicates, and then combines the outcome with $\boldsymbol{M}$

[2] Specifically, the *context* of a predicate in the body of an example is the head of that example.

[3] A mode declaration M_1 is inconsistent respect to a mode M_2 iff $M_1 \neq M_2$ (i.e., M_1 has input/output terms different from M_2). The internal representation of *Modes* in Algorithm 1 is a set $M=\{m_1,\dots,m_n\}$ where each m_i is the list of modes for the literals in the body of an example. A mode m_p for a predicate *p* is inconsistent with respect to the set M, if there is a list of modes $m_i \in M$ such that m_i contains a mode $m_{p'}$ for the predicate *p* inconsistent with respect to m_p.

in order to exclude inconsistent ones. Modes in ***M*** are grouped by concept, and represented as couples of the form "*m(h,b)*" where *h* is a concept along with one possible mode for its arguments (e.g., p(+,–)) and *b* is the list of all corresponding modes for the other predicates.

Note that, the restriction of having a single mode for each context can be relaxed in the algorithm, by just dropping the consistency test. This is useful when a predicate is used with different meanings. However, such cases may give rise to issues that need further discussion (outside the scope of this paper). Consider for instance the following predicate for reversing a list:

```
                         reverse([3,1,2],[2,1,3]).
```

represented by the example:

```
   rev(a,b):- decomp(a,3,c),decomp(c,1,d),decomp(d,2,e),nil(e),
              decomp(b,2,f),decomp(f,1,g),decomp(g,3,e).
```

where `a` represents the list `[3,1,2]` and `b` the list `[2,1,3]`. The following modes are obtained by applying the algorithm:

```
   {rev(+,-),decomp(+,-,-),decomp(-,+,-),decomp(-,+,+),
     decomp(+,-,+),nil(+)}
```

The *strange* mode is `decomp(-,+,-)` expressing that it is possible to obtain the tail of a generic list with a given head. The problem is due to one predicate (`decomp/3`) being used both to *decompose* (meaning represented by the mode `decomp(+,-,-)`) and to *recompose* (expressed by the mode `decomp(-,+,+)`) a list. One solution would be to exploit different predicate names for different uses. However, a more clever solution is based on the consideration that this mode is obtained from literal `decomp(g,3,e)`, which is applicable if and only if the third argument is a constant `e` such that `nil(e)` is true (i.e., the empty list).

4 Examples

In the outlined mode declarations learning task, it is possible to encounter problems due to the ambiguity of a predicate mode. In some cases, this ambiguity can be resolved by subsequent observations (if any) provided to the algorithm. In other cases, the ambiguity cannot be avoided since it depends on the context in which the predicate is used. The presented algorithm is able to recognize such cases and to manage them correctly. This section shows, by means of some examples, the capabilities of the algorithm in such different situations.

The internal representation of ***M*** is a structure of the following form:

```
  md([ [cm_1, [mp_11, mp_12, ... ] ],
       [cm_2, [mp_21, mp_22, ... ] ],
           ... ,
       [cm_n, [mp_n1, mp_n2, ... ] ] ).
```

where `md` represents ***M***, `cm`$_i$ represents a possible concept with one of its possible modes, and the `mp_ij`'s represent a possible list of modes for body predicates in the context of `cm_i`.

The first example shows as the incremental feature of the algorithm turns out to be important in some contexts.

Example 4. Given the example:

```
h(a):-p(a,b),p(a,c),t(a,b),t(a,c),l(c).
```

MILE induces the following list of modes for the literals `h/1`, `p/2`, `t/2` and `l/1`:

```
md([[h(+),[[l(+),t(+,+),p(+,-)],[l(+),t(+,-),p(+,+)]]]]).
```

According to the algorithm presented in Section 3, the argument `a` in the head leads to the following 1-layer literals: `p(a,b)`, `p(a,c)`, `t(a,b)`, `t(a,c)`. The first argument (`a`) of the predicates `p/2` and `t/2` is certainly an "input argument", but there is an indeterminism on the second argument, because both literals of `p/2` and literals of `t/2` produce simultaneously the same set of constants {`b,c`}. Since it is impossible to determine which of these two predicates actually gives as output the terms {`b,c`}, the system keeps both possibilities, hoping that a new incoming example resolves this ambiguity. Indeed, supposing that the next example provided to the system is

```
h(a) :- p(a,b),p(a,c),t(a,b),l(c).
```

MILE is able to induce the following list of modes for this example

```
md([[h(+),[[l(+),t(+,+),p(+,-)]]]])
```

that causes the following updated version of the global one (in which ambiguity is removed):

```
md([[h(+),[[l(+),t(+,+),p(+,-)]]]])
```

Hence, after examining just two examples, MILE learned the correct mode declarations for the predicates `h/1`, `l/1`, `t/2` and `p/2`. All new incoming examples must satisfy these declarations, or else an error will be notified. Indeed, the semantics that can be associated to these predicates is that `p/2` introduces new objects in the description, while `t/2` checks a relation between two known objects and, finally, `l/1` represents a property of the objects. Hence, supposing that the system is provided with the new example

```
h(a):-p(a,c),t(a,b),l(b).
```

it notes that the semantics (i.e., the mode declaration) of its predicates is different from that learned so far: *the predicate* `t/2` *now introduces objects in the description*. Then, the algorithm notify at the user that his descriptions associate different (or inconsistent) semantics to predicates. Note that this behavior can be avoided relaxing the consistency check.

The following example aims at showing a case in which the algorithm learns many mode declarations for one concept.

Example 5. Given the following example regarding the domain of family relationships

```
father(a,b) :- parent(a,b),male(a).
```

MILE is able to induce the following list of modes for the concept `father/2`:

```
md([ [ father(-,+), [ [male(+),parent(-,+)] ]],
     [ father(+,-), [ [male(+),parent(+,-)] ]],
     [ father(+,+), [ [male(+),parent(+,+)] ]]])
```

Due to a poor example description (only two literals in the body of the example), all the possible modes for the predicate `father/2` are correct (no inconsistency are derived). More in general, it is impossible to associate the correct semantics to the predicates `father/2` and `parent/2`. Supposing that MILE is provided with this second example, whose description is more detailed than the first one:

```
  father(a,b):-parent(a,b),male(a),parent(b,c),female(c).
```

Now, the algorithm is able to understand that the only correct mode for the predicate `father/2` is `(+,-)`:

```
  md([[father(+,-),[[female(+),male(+),parent(+,-)]]]])
```

Indeed, since `female(c)` is only used to test a property, the only predicate that can provide 'c' is `parent(b,c)`. Thus, `parent/2` must have mode `(+,-)`, and hence this leads to mode `(+,-)` for the predicate `father/2`.

Finally, the following example presents an important characteristic of the algorithm: its capability to learn many mode declarations for the same predicate when it is involved in different contexts.

Example 6. Supposing the following examples are provided to the algorithm:

```
  father(a,b) :-
    parent(a,b),male(a),parent(b,c),female(c).
  tree(t) :-
    node(t,a),node(t,b),node(t,c),parent(a,b),parent(a,c).
```

The user exploited the same predicate `parent/2` to describe both the family relationship concept `father/2` and the data structure concept `tree/1`. In this case, it is important to distinguish the two uses by linking the mode to its context. MILE induced the following list of modes:

```
  md([[tree(+),      [ [node(+,-),parent(+,+)] ]],
      [father(+,-), [ [female(+),male(+),parent(+,-)]]]])
```

The algorithm distinguished different semantics for the same predicate: in the concept (context) `tree/1` the predicate `parent/2` is used to check a property of the objects (*nodes*) introduced by the predicate `node/2`; while in the concept (context) `father/2` the predicate `parent/2` gives the second argument as output, thus allowing to 'retrieve' children of a given person.

The algorithm MILE is integrated as preprocessor in a system for the incremental learning of first-order logic theories from examples, called INTHELEX, that is included in the architecture of the EU project COLLATE[4], in order to learn rules for automated classification and understanding of paper documents [17]. A deeper analysis about the best way to exploit/integrate the meta-knowledge of mode declaration in the revision phases of INTHELEX is currently ongoing, in order to assess the gain obtained in terms of computational complexity.

[4] IST-1999-20882 project COLLATE: Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material (URL: http://www.collate.de).

5 Conclusions

One of the biases used by Inductive Logic Programming (ILP) systems for reducing the space of candidate solutions in knowledge acquisition and concept formation is mode declaration. An algorithm to incrementally learn mode declarations for predicates from examples, called MILE has been presented. Since such information can be useful for restricting the search space of ILP systems, implementations of the proposed algorithm could be profitably used as a preprocessor of input examples in ILP systems in order to provide them with this meta knowledge. At present, the algorithm is implemented in Prolog language. Important features of MILE are its ability to manage and resolve (whenever possible) ambiguity, and to capture different semantics for each predicate in relation with its use in different contexts. We plan to extend the algorithm in order to deal with predicates used with different meanings in the same context.

References

1. Eric McCreath and Arun Sharma. Extraction of Meta-Knowledge to Restrict the Hypothes Space for ILP Systems. Eight Australian Joint Conference on Artificial Intelligence,pp.75--82, Xin Yao, 1995
2. D.H.D. Warren. Implementing Prolog – Compiling Predicate Logic Programs. Research Reports 39 and 40, Dept. of Artificial Intelligence, University of Edinburgh, 1977.
3. C.S. Mellish. The Automatic Generation of Mode Declarations for Prolog Programs. DAI Research Paper 163, Dept. of Artificial Intelligence, University of Edinburgh, Aug. 1981.
4. C.S. Mellish. Some Global Optimizations for a Prolog Compiler. J. Logic Programming 2, 1 (Apr. 1985), pp.43-66.
5. U.S. Reddy. Transformation of Logic Programs into Functional Programs. In Proc. 1984 Int.Symposium on Logic Programming, IEEE Computer Society, Atlantic City, New Jersey, Feb. 1984, pp.187-196.
6. M. Bruynooghe, B. Demoen, A. Callebaut and G. Janssens. Abstract Interpretation: Towards the Global Optimization of Prolog Programs. In Proc. Fourth IEEE Symposium on Logic Programming, San Francisco, CA, Sep. 1987.
7. H. Manilla and E. Ukkonen. Flow Analysis of Prolog Programs. In Proc. Fourth IEEE Symposium on Logic Programming, San Francisco, CA, Sep. 1987.
8. S.K. Debray and D.S. Warren. Automatic Mode Inference for Logic Programs. Journal of Logic Programming, vol.5, n.3, pp.207-229, 1988.
9. K. Morik, S. Wrobel, J. Kietz, and W. Emde. Knowledge Acquisition and Machine Learning: Theory Methods and Applications. Academic Press, 1993.
10. S. Muggleton and C. Feng. Efficient Induction of Logic Programs. In Proceedings of the First Conference on Algorithmic Learning Theory, Tokyo, pp.368-381. Ohmsa Publishers, 1990.
11. R.M. Cameron-Jones and J.R. Quinlan. Efficient top-down induction of logic programs. SIGART Bulletin, 5(1):33-42, 1994.
12. S. Muggleton, Inverse Entailment and Progol. New Generation Computing, Special issue on Inductive Logic Programming, 13 (3-4), Ohmsha, pp.245-286, 1995.

13. M. Furusawa, N. Inuzuka, H. Seki and H. Itoh. Bottom-up induction of logic programs with more than one recursive clause. In Proceedings of IJCAI97 workshop Frontiers of ILP, Nagoya, 1997.
14. P. Idestam-Almquist. Efficient induction of recursive definitions by structural analysis of saturations. In L. De Raedt (Ed.), Advances in Inductive Logic Programming, pp.192-205. IOS Press, 1996.
15. E. McCreath and A. Sharma. LIME: A System for Learning Relations. Algorithmic Learning Theory, pp.336-374, 1998.
16. C. Rouveirol. Extensions of Inversion of Resolution Applied to Theory Completion. Inductive Logic Programming, pp.64--90, S. Muggleton, Academic Press, 1992.
17. F. Esposito, S. Ferilli, N. Fanizzi, T.M.A. Basile and N. Di Mauro. Incremental Multistrategy Learning for Document Processing. Applied Artificial Intelligence Journal, 17:859-883, Taylor & Francis, London, 2003.

TSP Optimisation Using Multi Tour Ants

Tim Hendtlass

Centre for Intelligent Systems and Complex Processes
School of Information Technology
Swinburne University of Technology
{thendtlass@swin.edu.au}

Abstract. Ant colony optimisation has proved useful for solving problems that can be cast in a path length minimisation form, particularly the travelling sales person (TSP) problem. Finding good, if not optimal, solutions in a reasonable time requires a balance to be struck between exploring new solutions and exploiting known information about possible solutions already examined. A new algorithm in which individual ants each live long enough to explore multiple solutions is introduced. Results are presented that show that enabling ants to learn from their own prior experience in addition to the collective wisdom of the colony improves performance on two standard test TSP data sets and suggests that the algorithm may well be useful for the whole class of TSP problems.

Keywords: Ant colony optimisation, travelling salesperson problem, heuristic search.

1 Introduction

Optimisation algorithms require a balance between exploration of new possibilities and the exploitation of prior experience. Excessive exploration can result in unsystematic exhaustive search while extreme excessive exploitation can result in endless re-evaluation of the previously explored results. Managing the balance between these two factors is a critical part of algorithm design. It is clear that algorithms exist which, even without explicit control of the ratio between these conflicting requirements, regularly find very good optima. These algorithms cannot guarantee to find the absolute optimum, but they find their (possibly local) minimum orders of magnitude faster than it would take an exhaustive search algorithm to find the global optimum.

Ant colony optimisation (ACO) algorithms are examples of such algorithms [DOR92,]. Each of these algorithms minimises the total cost of performing a fixed set of actions exactly once and returns an action order with a small (ideally the smallest) associated cost. It is most commonly described in the context of the travelling sales person (TSP) problem in which the actions are visits to locations and the cost is the total distance travelled, the sum of the path segments that join the various cities. If the length of the path segments is the same regardless of the direction of travel the resulting TSP is called a symmetrical, otherwise the TSP is called asymmetric. Many papers have solved the TSP using ACO methods including

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 523-532, 2004.

the seminal DOR96, together with DOR97, DOR97b, DOR99 and STU99. This last paper contains a summary of all ACO applications to the TSP to publication date.

The algorithms use an analogue of the chemical pheromone used by real ants. This pheromone is attractive to and tends to be followed by ants but is not the only factor influencing their path choices. Pheromone evaporates over time (diminishing the influence of old information) but is reinforced by ants laying more pheromone of each path they travel. The pheromone trails build up a collective estimation of the worth of the various possible path segments.

A simple description, applicable to the whole set of ant colony optimisation algorithms, is as follows. The individual algorithms differ in detail as described below.

1. The pheromone on every path segment is initialised to an initial value (τ_0) and N ants are randomly distributed among C cities. This initial city, although part of the path, is not considered as having been move to.
2. Each ant then decides which city to move to next. The exact way they do this is algorithm dependent and described later, however, ants are always prohibited from returning to any city previously moved to. Each ant moves to its chosen city and then considers the city it should move to next, repeating this process until it is back at the city it started from. At this point the ant has completed one tour.
3. Each ant calculates the length of its tour, and updates the information about the best (shortest) tour found so far if necessary.
4. The pheromone levels on each path segment are then updated, the exact update strategy is algorithm dependent but all involve evaporating pheromone and adding extra to selected path segments.
5. All ants that have completed their maximum number of tours (often one) die and are replaced by new ants at a randomly chosen cities. No change is made to pheromone levels.
6. The algorithm continues from step 2 and the process continues until some stopping criteria is met, such as the best path being below some threshold or a maximum total number of tours having been completed.

It is in the detail of steps 2 and 4 that the various algorithms differ. Two of the most common algorithms, AS and ACS, are described below together with the detail of the AMTS algorithm introduced in this communication.

1.1 Ant Systems (AS)

The original of these algorithms, the Ant System (AS) algorithm was inspired by the behaviour of Argentinean ants [DOR92].

In AS an ant at city i calculates which city to visit next by first calculating the attractiveness A_{ij} of each possible city j. $A_{ij} = 0$ if the city has been visited before or $A_{ij} = P_{ij}^{\alpha}\ D_{ij}^{\beta}$ if it has not. P_{ij} is the pheromone level on the path segment from city i to city j, D_{ij} is the distance from city i to city j and α and β are two user chosen parameters, typically 2 and –2 respectively. The probability that this ant moves from city *i* to city *j* is $\frac{A_{ij}}{\sum_{j=1}^{C} A_{ij}}$ and the decision as to which city to visit next is solely dependent on this set of probabilities.

In AS the pheromone levels are updated simultaneously once all ants have completed their tour. The pheromone on every path segment is first decreased by being multiplied by $(1-\rho)$, where ρ is a parameter less than unity. Each ant then augments the pheromone on all the path segment it used as part of its tour by adding ρ_u / L units of pheromone to that path segment, where ρ_u is another parameter and *L* is the length of that ants tour.

Deciding which city to visit next from the set of probabilities involves at least one random numbers at it is this that introduces the exploration element into the algorithm. However, as the balance between exploration and exploitation is stochastic, it cannot be accurately predicted.

1.2 Ant Colony System (ACS)

The Ant Colony System (ACS) [DOR99] differs from AS in two important ways; the method of deciding the next city to visit and the pheromone update procedure. These changes make the algorithm more complex but generally lead to better performance than AS.

The decision process starts with the calculation of the attractiveness A_{ij} of each city *j* using the same formula as for AS except that the value of α is unity. A random number is then compared to a preset threshold[1], the greedy probability. If the random number is below this threshold the city with the highest attractiveness is chosen, otherwise the same probabilistic selection process is used as in AS. This change increases the emphasis on exploitation compared with AS and has the effect of ensuring that many of the path segments each ant follows are in common with the currently best known path, thus encouraging such exploration as occurs to occur around this best known path.

The second change is that two different types of pheromone update occur. The best-found path is reinforced even if no ant has exactly followed that path this tour. This update mechanism does not alter the pheromone levels on any path segment not involved in the best-known path. The pheromone on every path segment of the best known path is first decreased by being multiplied by $(1-\rho)$ where ρ is a user chosen parameter (less than unity) and then has ρ_u / L^{+} units of pheromone added to it, where L^{+} is the length of this best known path.

[1] Actually there is no reason why the threshold should not be modified as the algorithm progresses. While such an approach is attractive it has not been done in this work as the algorithm for varying the threshold introduces yet another function with the potential to mask the effect of the algorithm variation described for the first time in this communication.

The second update mechanism updates the pheromone on every path segment actually travelled by an ant, but in such a way as to *reduce* the value of the pheromone on these segments. Each ant updates all the path segments it used by first multiplying the current pheromone level by $(1-\rho)$ and then adding $\rho * \tau_0$ units of pheromone where τ_0 is the initial value of pheromone deposited on all path segments.

The pheromone levels on used segments that are not part of the current best path are moved asymptotically down towards $\rho * \tau_0$ while the path segments used in the currently best known path have their pheromone levels move asymptotically up. Note that the pheromone levels on path segments neither involved in the best-known path nor travelled by any ant this tour are left unchanged.

As a result, ants that do not travel on a best path segment are more likely - in time - to choose an unused (or underused) segment. These two pheromone updates simultaneously encourage exploration of unused path segments and the exploitation of the segments that are part of the current best-known path.

1.3 Max-Min Ant System

If the pheromone level on the most densely travelled path becomes high, many ants will take this path further reinforcing the high pheromone level. This will have the effect of decreasing the amount of exploration. At the same time, the pheromone levels on infrequently travelled path segments can drop so as to be insignificant, again decreasing the probability that they will ever be explored. Just normalising the pheromone densities on the paths to and from a city does not help as this does not alter the relative levels on the frequently and infrequently travelled path segments.

One way of addressing these problems is to limit both the maximum and minimum permissible pheromone levels. One algorithm for doing this is the Max-Min algorithm [STU00]. In this algorithm the pheromone on all path segments is initialised to the maximum density that is to be allowed, $\tau_{\max}$, which is equal to 1/ρ where ρ is the evaporation rate. When all ants have completed their tour, all paths suffer evaporation during which their pheromone is reduced by a factor of (1- ρ). A minimum pheromone level is specified for any path segment ($\tau_{\min}$). Any path segment whose pheromone level would drop below this as a result of the application of the (1- ρ) factor has its pheromone level reset to $\tau_{\min}$. The path segments that make up the best path (only) have pheromone added to them. One unit of pheromone is added to their pheromone level but the level is then reduced to $\tau_{\max}$ if necessary.

1.4 Ant Multi-tour System (AMTS)

Both AS and ACS share one particular common feature - after an ant has completed a tour it dies and is replaced by a new ant at some randomly chosen city. In the Ant Multi-Tour System (AMTS) ants do not die until they have completed a used chosen number of tours. After their first tour each ant carries some information from its previous tours; in particular how many times it has previously moved from city *i* to city *j*. This information is used to encourage the ant not to (exactly) follow the path that it travelled previously but to explore variations to these paths.

Each ant builds up a list of the prior path segments it has followed from each city. It used this information to reduce the probability of repeating a choice previously made.

The only difference to the ant system algorithm is this modification made to the probability of an ant moving from city *i* to city *j*, which is now calculated using $\frac{A_{ij}}{F_{ij}\sum_{j=1}^{C} A_{ij}}$. In this calculation the extra term F_{ij} on the denominator is a factor that reflects the number of times that this ant has previously chosen to move from city *i* to city *j* in previous tours ($prior_{ij}$). In the work reported in this communication, $F_{ij} = 1 + \sqrt{prior_{ij}}$. As $prior_{ij}$ only reflects a singular transition, the storage requirements are trivial compared to storing all parts explored.

Reducing the probability of remaking choices that have already been made by this ant encourages each ant to increase exploration at the cost of exploitation on all tours after its first.

2 The Burma 14 Data Set

The Burma 14 data set shown in Table 1 is a relative small symmetric TSP problem. However, it is large enough to test optimisation TSP algorithms while simultaneously being small enough that the full set of possible solutions can be exhaustively generated. The problem is to find the shortest tour between the 14 cities, specified by their latitude and longitude in the data set, with each city being visited exactly once. There are 14 possible starting points for each closed tour and each tour can be traveled in two different directions.

Table 1. The Burma 14 data set.

City	Latitude	Longitude	City	Latitude	Longitude
0	16.47	96.10	7	17.20	96.29
1	16.47	94.44	8	16.30	97.38
2	20.09	92.54	9	14.05	98.12
3	22.39	93.37	10	16.53	97.38
4	25.23	97.24	11	21.52	95.59
5	22.00	96.05	12	19.41	97.13
6	20.47	97.02	13	20.09	94.55

None of these choices affect the tour length. The number of different tours is therefore 13!/2, or approximately $3*10^9$.

The length $dist_{ij}$ of the path segment between any two cities *i* and *j* (latitudes la_i, la_j and longitudes lo_i, lo_j respectively) is given by the following calculation.

$dist_{ij} = \mathrm{int}(6378.388 * a\cos(0.5*((1+q1)*q2) - ((1-q1)*q3)+1)$ where

$$q1 = \cos(lo_i - lo_j) \quad q2 = \cos(la_i - la_j) \quad q3 = \cos(la_i + la_j)$$

$$lat1 = \pi(\text{int}(la_i) + 5(la_1 - \text{int}(la_i)) / 3) / 180$$

$$lat2 = \pi(\text{int}(la_j) + 5(la_2 - \text{int}(la_j)) / 3 / 180$$

$$long1 = \pi(\text{int}(lo_i) + 5(lo_1 - \text{int}(lo_i)) / 3 / 180 \textit{ and}$$

$$long2 = \pi(\text{int}(lo_j) + 5(lo_2 - \text{int}(lo_j)) / 3 / 180$$

Note that the length of each path segment is converted to an integer as prescribed in the TSLIB library of TSP problems [REI95].

Table 2. The top ten paths for Burma14.

Length	City order		Length	City order
3323	0,1,13,2,3,4,5,11,6,12,7,10,8,9		**3359**	0,1,13,2,3,4,11,5,6,12,10,8,9,7
3336	0,1,13,2,3,4,5,11,6,12,10,8,9,7		**3369**	0,7,1,13,2,3,4,11,5,6,12,10,8,9
3346	0,1,13,2,3,4,11,5,6,12,7,10,8,9		**3371**	0,7,12,6,11,5,4,3,2,13,1,9,8,10
3346	0,7,1,13,2,3,4,5,11,6,12,10,8,9		**3381**	0,1,13,2,3,4,5,11,6,12,7,9,8,10
3359	0,1,13,2,3,4,5,11,6,12,10,9,8,7		**3381**	0,1,13,2,3,4,5,11,6,12,8,9,10,7

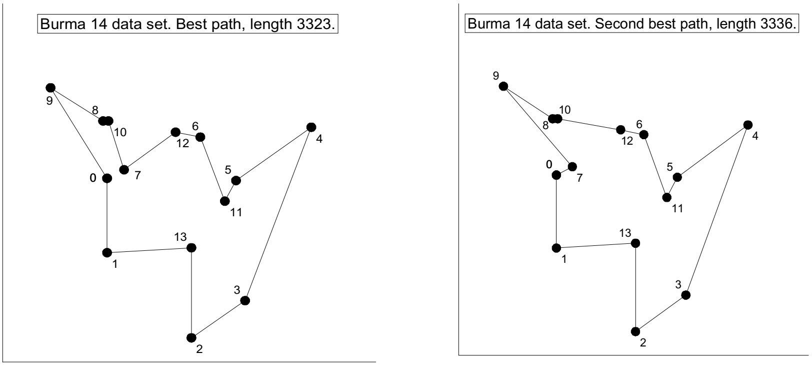

Burma 14 data set. Best path, length 3323.
9
8
10
0
7
12
6
5
4
11
13
1
3
2
Burma 14 data set. Second best path, length 3336.
9
10
8
0
7
6
12
5
4
11
13
1
3
2

Fig. 1. The two best paths for the Burma 14 data set.

Table 3. The number of path segments different between the top 10 paths for Burma 14.

3336	3346	3346	3359	3359	3369	3371	3381	3381	
3	2	3	5	3	6	5	2	3	**3323**
	5	3	2	2	4	4	2	3	**3336**
		3	6	3	3	5	4	5	**3346**
			4	5	2	3	4	5	**3346**
				2	6	5	3	2	**3359**
					3	6	4	5	**3359**
						5	6	7	**3369**
							2	5	**3371**
								4	**3381**

The length and the actual city orders for the ten shortest tours are given in Table 2 with the two shortest tours shown in Fig. 1.

Table 3 lists the number of path segments different between the top ten paths.

3 Results

Each of the three algorithms (AS, AMTS and ACS) was used to solve the Burma 14 TSP problem using the parameters shown in Table 4. The AS and AMTS values had been previously found by the author to be effective for the Burma 14 data set, the ACS values follow the recommendations in [DOR99]. An iteration consists of all ants completing one tour. Runs were terminated after 100 iterations had occurred without a new best path being found.

Table 4. Parameter values used for each of the algorithms for Burma14.

	Ant count	Alpha	Beta	Pheromone					Greedy probability
				Initial	Decay	Update	Max	Min	
AS & AMTS	10	2	-2	0.1	0.1	0.5	n.a.	n.a.	n.a.
ACS	10	1	-2	$2.2*10^{-5}$	0.1	0.1	n.a.	n.a.	0.9
Max Min	10	1	-2	3.3	0.3	1	3.3	0.1	n.a

Table 5. AS, ACS and AMTS results. All results expressed as a percentage.

Algorithm	AS	AMTS $F_{ij} = 1 + \sqrt{prior_{ij}}$						ACS	Min Max
Max age	na	2	3	4	5	6	7	na	na
Path length = 3323	6	19	50	53	56	59	40	15	17
Path length = 3336	41	32	32	42	38	39	59	53	62
Path length = 3346	7	3	5		2				
Path length = 3359	15	16	10	3	2				
Path length = 3369	1	1							
Path length = 3371	4	8				1		22	10
Path length = 3381	23	20	3	2	2	1	1	9	6
Path length > 3381	3	1						1	5
Average excess path	30.29	27.16	10.65	7.7	7.28	6.13	8.25	24.08	22.09
Average iterations	47.15	51.51	48.6	41.49	38.4	29.99	22.02	9.45	30.54

The results are shown in Table 5. The average excess path length is the difference between the average path found and the best possible path length of 3323.

4 Discussion

It is clear that the AMTS algorithm outperforms both of the other algorithms on the Burma 14 TSP, both in the quality of the solutions that it finds and also the time it takes to find them. The parameters used for the ACS algorithm are the generic values suggested by [DOR99] while the parameter values used for AS have been optimised for this data set during prior work. This may account for the relatively poor results obtained using the ACS algorithm, which while tending to find poorer solutions found them faster than AS.

The AMTS algorithm typically ends at one of the two shortest paths. Consider

Table 3 that shows the number of path segments that differ between each of the top ten paths. It can be seen that transitions from the majority of the poorer paths to the second best path involves one or more steps that require a generally smaller number of path segments to be changed than to move to the best path. Moving directly from the second best to the best path requires three path segments to change with the result that both the two best paths of the Burma 14 TSP data set are fairly stable, as reflected in the percentage of times that one of them is the final solution found.

As the maximum age of the ants increases in the AMTS algorithm a point is reached at which the quality of the solution starts to degrade, even though the speed with which they are found continues to fall. This happens as the value of F_{ij} becomes so large that the ant rarely follows any path segment of the current best path. Since it is likely any better path will share at least some (and probably many) path segments with the current best path, such a strong disinclination becomes counter productive.

The ACS and AMTS algorithms have a closer similarity in intent and approach than might at first appear. The key difference is that ACS works on a global level with the best path found by anyone in the whole colony being reinforced and all path segments travelled by any ant being made less attractive to all other ants. In AMTS each ant reinforces the best path it has personally found so far while each ant is only influenced not to re-explore the path segments it personally has travelled in the past. This leads to a greater diversity of collective ‘wisdom’ about good path segments in AMTS and reduces the emphasis on exploitation compared to ACS.

This ant specific nature of the re-exploration deterrent in AMTS, rather than the global deterrent of ACS, seems more appropriate. Each ant must evaluate the undesirability of each possible path segment based on the experience of and the paths already tried by this ant rather than on the net experience gained from all the paths travelled before by any ant. The ACS method, while useful, is unlikely to be as useful and thus the targeting of the exploration in AMTS is likely to be better. Indeed since AMTS always involves a probabilistic decision at each city, as opposed to often taking the most path with the highest desirability as in ACS, the quantity of exploration may be expected to be higher too.

For a TSP a new good path will usually include many path segments in common with the previously best-known path. An example of this can be seen from Figure 1 in which the difference between the best and second best path is restricted to the path segments between the cities in the top left corner of the plots. As a result, an over emphasis on exploration becomes counter productive as all paths tried are likely to contain too few of the already discovered necessary path segments. Thus there is likely to be an upper limit to the useful age of, and therefore the number of tours completed by, an ant. For the work described here the useful upper age limit appears to be about six. The useful upper age limit is almost certainly problem dependent.

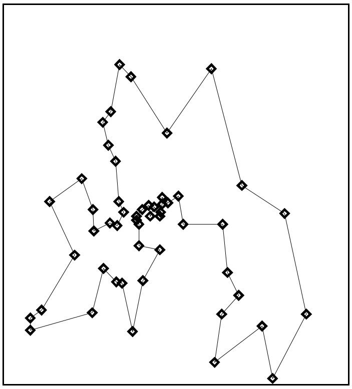


Fig. 2. The best solution for the Berlin 52 city TSP data set.

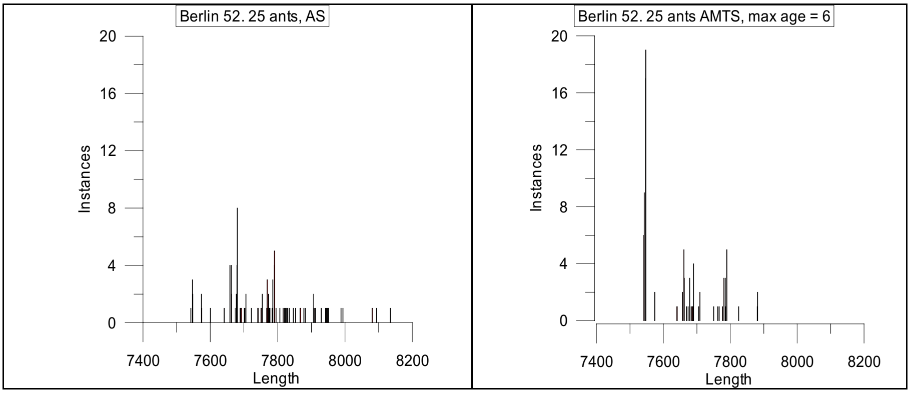


Fig. 3. The results of 100 repeats of the AS algorithm (left) and the AMTS algorithm (right) on the Berlin 52 TSP data set.

Table 6. The relative performance of the AS and AMTS algorithms on the Berlin 52 TSP data set.

	AS	AMTS max age 6
Average excess path	216.4	90.8
Average iterations	66.0	33.1

The results for the Burma 14 data set, while significant, are not in themselves conclusive as there remains the possibility that it is some aspect of this particular data set that responds to the AMTS algorithm better than to the AS algorithm. Preliminary results (shown in figure 3) have also been obtained for the Berlin 52 data set with approximately $7*10^{65}$ paths. The best solution is shown in Fig. 2. It should be noted that, apart from increasing the number of ants to 25, all the parameters used for both the AS and AMTS are identical to the values used for the Burma 14 data set and are unlikely to be optimum. It is not the absolute results that are important here, although the best solution (with a path length of 7542) was found (once by AS and nine times by AMTS in a 100 repeats). What is important is the improvement that accrues from allowing the ants to take multiple tours. As can be seen from the results quoted in this paper, this is significant with the AMTS algorithm finding better solutions considerable faster. Overall the results presented in this paper suggest that the AMTS algorithm may well be beneficial for TSP problems in general.

5 Conclusions

Allowing ants to have access to both the collective wisdom of their colony and their own prior life experience has been shown to be beneficial when solving at least two TSP data sets. Further work will be needed to truly evaluate the relative performances of AMTS and ACS.

At this stage of development ants do not share their ‘life experiences’ when they meet at a city. Nor is the influence of the prior experience factor a function of the number of times that an ant has already departed from their personal best path. These refinements may be implemented and tested in future work.

References

[DOR92] Dorigo, M. (1992) Optimization, Learning and Natural Algorithms, PhD Thesis, Dipartimento di Elettronica, Politechico di Milano, Italy.

[DOR96] Dorigo, M., Maniezzo, V. and Colorni, A. (1996) ``The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man and Cybernetics - Part B, 26, pp. 29-41.

[DOR97] Dorigo, M. and Gambardella, L. (1997) ``Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computing, 1, pp. 53-66.

[DOR97b] Dorigo, M and Gambardella, L. (1997) ``Ant Colonies for the Traveling Salesman Problem", Biosystems, 43, pp. 73-81.

[DOR99] Drigo, M. and Di Caro, G. (1999) ``The Ant Colony Optimization Meta-heuristic", in New Ideas in Optimization, Corne, D., Dorigo, M. and Golver, F. (eds), McGraw-Hill, pp. 11-32.

[REI95] Rinelt G. TSPLIB95. Available from: http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB95.html

[STU99] tutzle, T. and Dorigo, M. (1999) ``ACO Algorithms for the Traveling Salesman Problem", in Evolutionary Algorithms in Engineering and Computer Science, Miettinen, K., Makela, M., Neittaanmaki, P. and Periaux, J. (eds), Wiley.

[STU00] Stützle T and Hoos H.H.. Max-Min Ant Systems. Future Generation Computer Systems, 16(8):889-914, (2000)

Neighborhood Selection by Probabilistic Filtering for Load Balancing in Production Scheduling[1]

Byoungho Kang and Kwang Ryel Ryu

Department of Computer Engineering, Pusan National University,
Jangjeon-Dong San 30, Kumjeong-Ku 609-735, Busan, Korea
{bhokang, krryu}@pusan.ac.kr

Abstract. For a local search algorithm to find a better quality solution it is required to generate and evaluate a sufficiently large number of neighbor candidate solutions at each iteration, demanding quite an amount of CPU time. This paper presents a method of selectively generating only good-looking neighbors, so that the number of neighbors can be kept low for efficient search. In our method, a newly generated candidate solution is probabilistically selected to become a neighbor based on a quality estimation made by a simple heuristic. Experimental results on the problem of load balancing in production scheduling have shown that our neighbor selection method based on probabilistic filtering outperforms other random or greedy selection methods in terms of solution quality given the same amount of CPU time.

1 Introduction

It is inevitable to use a heuristic search algorithm when given a large-scaled optimization problem. Popular heuristic search algorithms such as simulated annealing [1] and tabu search [2] start from a certain candidate solution and probe its neighborhood to find ones with improved quality. The search continues iteratively from point to point in the search space until a satisfactory solution is found. For this reason, these search algorithms are often called iterative improvement search or local search algorithms. To increase the possibility of obtaining a good solution by local search such as tabu search, we have to generate and evaluate enough number of neighbors for enough number of iterations. However, since the time taken for evaluation is proportional to the number of neighbors generated, we commonly generate only a certain limited number of neighbors randomly to keep the number of iterations sufficiently large. In an effort to make the search more efficient with limited number of neighbors, we sometimes generate neighbors selectively based on a greedy heuristic, taking the risk of prematurely converging to a local optimum.

This paper presents a neighbor selection method which is based on simple preliminary evaluation of the generated candidate solutions. Our method selects neighbors

[1] This work was supported by National Research Laboratory Program (Contract Number : M10203000028-02J0000-01510) of KISTEP.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 533-542, 2004.

based on what we call probabilistic filtering in the following way. First, a candidate solution is randomly generated and a preliminary evaluation is done using a simple heuristic measure. Then, a bias function is applied to map the evaluation value to a probability. Based on this probability, it is determined whether the candidate solution should become a neighbor or not. This process repeats until a predetermined number of neighbors are collected. Note that a precondition for our neighbor selection method to be meaningful is that the cost of preliminary evaluation should be ignorable compared to that of the original objective function evaluation. The bias function above is adjustable so that the resulting probability makes the search to become either more random or greedier depending on how much we give bias towards the results of preliminary evaluation.

Simulated annealing also employs probabilistic decision making in determining whether to move to a newly generated neighbor or not. The probability is derived from evaluation value through a bias function made up from an analogy with a physical process of annealing. However, the evaluation is done by using the original objective function of the given problem. As a matter of fact, we can apply our method to simulated annealing by applying the filtering process explained above as a preprocessing to select a neighbor before it is fully evaluated. As long as the preliminary evaluation is very cheap, the filtering as a preprocessing contributes to improving the search efficiency.

There are other previous works using probabilistic decision making in searching for solutions. The mixed random walk strategy by Selman and Kautz [3] solves hard satisfiability problems by using both random walk strategy and greedy local search, in an effort to improve the previous method called GSAT [4]. At each iteration with probability p, a variable is selected randomly from unsatisfied clauses and its value is changed, while with probability $(1 - p)$ greedy local search is conducted. However, there is no notion of generating and filtering neighbor solutions. Heuristic-biased stochastic sampling (HBSS) [5] improves iterative sampling method [6] [7] by applying a bias function to heuristic evaluation value to derive probability for selecting a node in constructive search. The original iterative sampling method finds a candidate solution, which is a path from the root to a leaf of a search tree, by repeatedly selecting a child node randomly. When a predetermined number of such solutions are collected the best one is returned, and this process repeats until a satisfactory solution is found. Instead of random sampling of a child node, HBSS heuristically evaluates all the child nodes and applies a bias function to assign probabilities based on which one child node is sampled. More recently, Binato et. al. [8] applied the idea of probabilistic job selection to solve job shop scheduling problems within the framework of GRASP [9]. GRASP is an iterative improvement search procedure and each of its iterations consists of two phases: construction and local search. In the work of Binato et. al., the construction phase builds a feasible job shop schedule and local search explores its neighborhood. It is in this construction phase where the probabilistic selection of a job is made. Therefore, their method can be viewed as an instance or a close variant of HBSS.

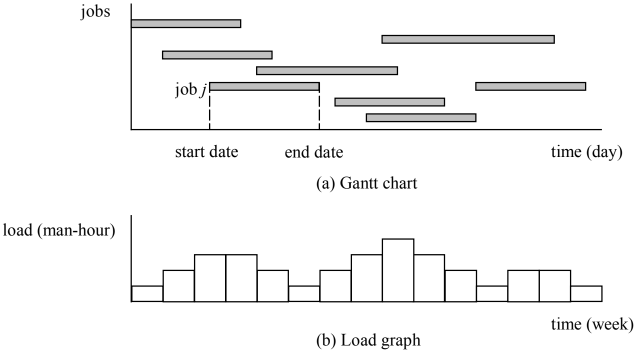
Our method presented in this paper can be considered an extension of HBSS to iterative improvement search. Our method is different from HBSS in that our method does a simple preliminary evaluation on a complete candidate solution, while HBSS does the evaluation using the original objective function but on a partial solution because it works within the framework of constructive search. Another difference is that our method is typically applied to collect many good-looking neighbor solutions, while HBSS is used to sample a single child node in the search tree.

We have tested both tabu search and simulated annealing with our neighbor selection method by applying them to a large-scaled real world problem of load balancing in production scheduling. Experimental results have shown that neighbor selection by probabilistic filtering outperforms other random or greedy selection in terms of solution quality given the same amount of CPU time. We have also been confirmed that the bias function, if adjusted appropriately, plays an important role in making the search efficient by providing a good compromise between exploration and exploitation.

The next section introduces the target problem of load balancing in production scheduling. Section 3 describes how we apply our method of neighbor selection by probabilistic filtering to the load balancing problem. Section 4 reports experimental results, and finally section 5 gives conclusions.

2 Load Balancing in Production Scheduling Problem

Production scheduling is the problem of allocating resources to the activities of multiple processes over time to achieve a targeted global behavior, while satisfying various constraints among the activities and the resources. In our target problem, we are given a long-term project which consists of many jobs to be done in some particular shops. The objective is to schedule the jobs in such a way that weekly loads of the shops are balanced as much as possible without violating given constraints. Figure 1(a) illustrates a job schedule of a particular shop in the form of a Gantt chart. A bar in the chart shows that the corresponding job is assigned to this shop and scheduled to start and end as indicated. Figure 1(b) shows the weekly accumulated load of all the jobs assigned to the shop. Both the load and the weekly capacity are measured in man-hours.



(a) Gantt chart

(b) Load graph

Fig. 1. Gantt chart of a shop schedule and its load graph

The objective of our scheduling problem is to adjust the start and end dates of each job so that the load of each of the shops is balanced as much as possible among the different weeks over the whole project term. The objective function is therefore

$$\min\left(\sum_{k=1}^{g} w_k \varepsilon_k^2\right) \qquad (1)$$

where g is the number of the shops, w_k is the weight value reflecting the importance of shop k, and ε_k^2 is the mean squared error of the weekly normalized loads of shop k with respect to the full weekly capacities, i.e.,

$$\varepsilon_k^2 = \frac{1}{t}\sum_{i=1}^{t}\left(\frac{L_{ki}}{C_{ki}} - 1\right)^2 \qquad (2)$$

where t is the number of weeks in the whole project term, L_{ki} is the accumulated load of shop k on the i-th week, and C_{ki} is the capacity of shop k on the i-th week.

The adjustment of start and end dates of each job, however, must be done carefully considering various constraints. The first such constraint is the partial ordering constraint; a job (or jobs) can start only after all the jobs constrained to precede it ended, and sometimes there can be a certain buffer period required during which the succeeding jobs should remain idle before they can start. Although each of the jobs has a predetermined shop where the job is to be done, the ordering constraint may exist among the jobs of different shops, making the load balancing problem more complicated. Another constraint to be noted is that each job has its own minimum and maximum possible durations. Therefore, the number of different ways of making adjustment of the start and end dates is limited. If there are n different jobs in a given project and m different ways of adjustment on the average, the size of the search space becomes m^n. In case of the problem we used for our empirical study, n is 467 and m is 214.

3 Neighbor Selection by Probabilistic Filtering

The load balancing problem described above can be solved efficiently by a local search employing our neighbor selection. The method consists of two stages by following the so-called successive filtration strategy [2] to reduce the amount of search; a candidate neighbor solution (a new schedule) is generated from the current solution (the current schedule) first by selecting a critical job and then by adjusting its start and end dates. A job is considered critical if the adjustment of its schedule is expected to contribute a lot to the improvement of the overall load balance.

In the first stage, a job belonging to a shop is picked up randomly and its criticality is estimated through a simple preliminary evaluation. Based on this evaluation, a decision is made probabilistically to determine whether the job at hand should be held as a critical job or not. If discarded, this random job picking and probabilistic filtering repeat until one is held as critical. Then in the second stage, it is determined whether the schedule of the critical job should be adjusted to move forward or backward along the time axis. This decision is also made probabilistically based on another simple heuristic evaluation. After the direction of adjustment is decided, the amount of adjustment is determined purely randomly because it is hard to guess how much adjustment is good for eventual load balancing. If a job schedule is to be adjusted to move forward, for example, the current start date and the following dates lying within the range allowed by afore-mentioned constraints are all equally likely to be picked up as a new start date. In succession, the end date is also changed in the same manner. The schedule obtained this way, however, may have no change on both start and end dates. When this happens, either the start or end date is adjusted by one day in the decided direction to make sure the new schedule is different from the old one. This two-stage process is iterated until an appropriate number of neighbors are collected. At each iteration, the jobs already selected as critical ones before are excluded from consideration. When only a single neighbor is needed as is the case with simulated annealing, the process terminates as soon as one neighbor is selected.

To estimate the criticality of a job it is helpful to have the Gantt chart overlapped with the load graph as shown in Figure 2. It is reasonable to guess that a job is critical if the load within the duration of the job is far deviated from the average load. By adjusting such a job we expect a large contribution to improving the load balance. Suppose job j is assigned to shop s. Let $d_{\max,s}$ be the maximum deviation of the load of shop s from its average load, and $d_{j,s}$ be the local peak deviation of the load of shop s within the duration of job j. Both $d_{\max,s}$ and $d_{j,s}$ can be obtained from either a peak or a valley. In any case the deviation is measured in absolute value. Then, the criticality value v_j of job j is simply calculated by $v_j = d_{j,s} / d_{\max,s}$.

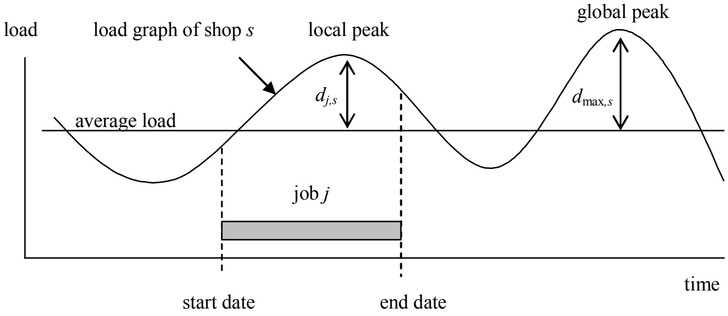


Fig. 2. Estimation of job criticality by comparing the peak loads of a shop.

To convert the criticality value v of a job to probability p_c of holding it as a critical job, the following bias function B_ρ is introduced:

$$p_c = B_\rho(v) = e^{-(1-v)/\rho}, \quad 0 \le v \le 1, \rho > 0 \tag{3}$$

The purpose of introducing the bias function B_ρ is to have control over the resulting probability distribution by adjusting the randomization factor ρ depending on the bias towards the criticality value calculated heuristically. When ρ is very large, $p_c \approx 1$ regardless of the value of v as can be seen in Figure 3. In this case a job is almost randomly selected as a critical job. As ρ becomes smaller, the selection becomes greedier. The criticality value itself can directly be used as probability also, which case corresponds to having no bias function (or equivalently an identity bias function, i.e., $p_c = v$). Various bias functions with different randomization factor ρ are shown in Figure 3.

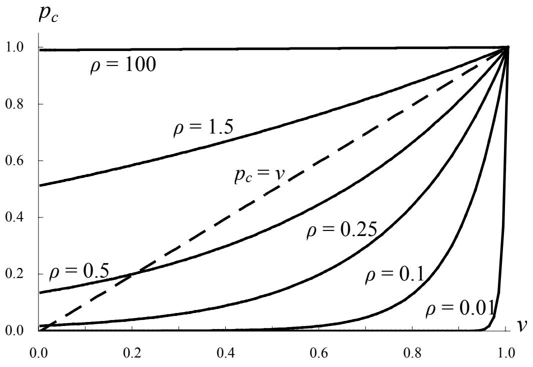


Fig. 3. Various bias functions with different randomization factor ρ.

After a critical job is selected, the direction of its schedule adjustment along the time axis must be decided. Figure 4 illustrates how we derive probability for deciding the direction of adjustment by observing the load graph over the duration of the critical job. Assuming that job j belongs to shop s, the figure shows that the load of shop s on the end date of job j is higher by $\delta_{j,s}$ compared to that on the start date. If τ_j is the duration of job j, the angle of gradient can be calculated by $\theta_{j,s} = \text{atan}(\delta_{j,s} / \tau_j)$, where $-\pi/2 < \theta_{j,s} < \pi/2$. We get a negative angle value when $\delta_{j,s}$ is negative. When $\theta_{j,s}$ is positive, job j might have to be moved in the backward direction along the time axis, i.e., its schedule might have to be adjusted to some earlier time. The reason is that we may expect some reduction of load around the end date of job j. With bigger value of $\theta_{j,s}$, we expect bigger effect of reducing the load by backward schedule adjustment. Therefore, the probability $p_b(j)$ of making backward adjustment of job j is made proportional to the value of $\theta_{j,s}$.

Given a critical job, the decision for the direction of schedule adjustment is made to be either backward or forward, with probability p_b for backward and accordingly $(1 - p_b)$ for forward. Once the direction is decided, the amount of adjustment in that direction is made purely randomly as described earlier in this section, thus producing a complete neighbor candidate solution. Note that p_b must be 1/2 when $\theta = 0$ because it is not clear which direction is the right one. When $\theta < 0$, p_b must be less than 1/2 because in that case the adjustment should better be made in the forward direction. Considering all these, p_b is derived from θ as follows:

$$p_b = \frac{1}{2} + \frac{\theta}{\pi} \tag{4}$$

Nonlinear bias function of the kind used before is not introduced here to map the angle value to probability. This decision has been made based on our empirical study. Note that the computation costs of equations (3) and (4) are ignorable compared to those of equations (1) and (2). This implies that the overhead of neighbor selection by probabilistic filtering is not of much concern.

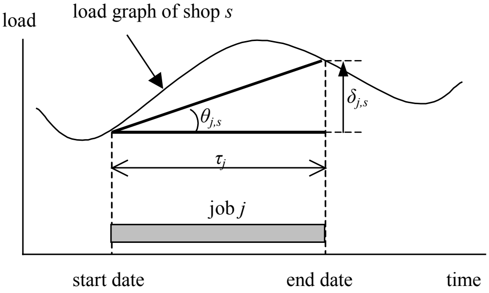


Fig. 4. The angle used for deciding the direction of schedule adjustment.

4 Experimental Results

We have conducted experiments for production scheduling with real data of a one-year project. The project consists of 467 jobs each of which should be done in one of the two specialized shops. The average number of different ways of schedule adjustment per job is 214. Given an initial schedule made by a human expert (an initial schedule can also be generated automatically by following some simple rules considering the given constraints), the task is to improve the balance of loads of both shops over the whole project term by adjusting the start and end dates of certain jobs while satisfying all the given constraints. This task is attacked by applying local search algorithms equipped with our neighbor selection method. We tested both tabu search and simulated annealing as our local search algorithm. The number of neighbors collected by probabilistic filtering is 23 (empirically determined) for tabu search and 1 for simulated annealing.

To see the effect of neighbor selection by probabilistic filtering, we compare the following three strategies for generating neighbor candidate solutions.

- **Random:** A job is randomly selected and its schedule is also randomly adjusted.
- **Greedy:** Jobs are selected in decreasing order of their local peak's or valley's deviation from the average load. The direction of schedule adjustment is decided to be backward when the gradient angle $\theta > 0$, forward when $\theta < 0$, and random when $\theta = 0$. The amount of adjustment is randomly determined.
- **NSPF:** Jobs are selected by probabilistic filtering with or without the bias function. The direction of schedule adjustment is decided probabilistically but the amount of adjustment is determined randomly.

Figure 5 shows typical runs of tabu search with various neighbor selection strategies given an initial schedule with the evaluation value of 18.9. The evaluation values are obtained by calculating equation (1). Greedy selection turns out to be much worse than random selection, and the winner is NSPF employing the bias function. The greedy selection prefers jobs which look the most critical. However, schedule adjustment of such jobs sometimes cannot be done by an enough amount due to the ordering constraints with other preceding or succeeding jobs. Once this schedule becomes the next candidate solution to move to anyway, it is likely that the search is caught in a local optimum hard to escape from. On the other hand, NSPF is more likely to select and adjust the schedules of less critical jobs. This sometimes provides enough room for other preceding or succeeding critical jobs to be adjusted into the right direction in subsequent iterations, thus giving opportunities for significant improvement. Note also that the nonlinear bias function plays an important role in making the search even more efficient. With an appropriate randomization factor ρ, the bias function seems to provide a good compromise between intensification and diversification of search.

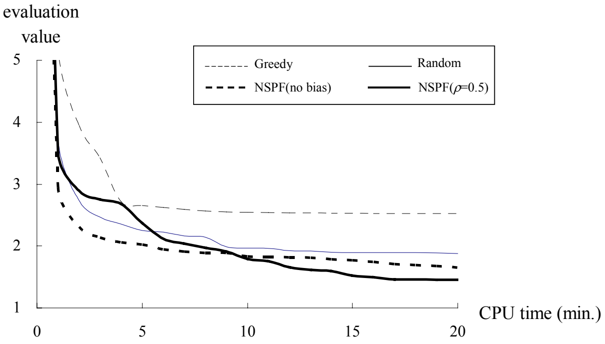


Fig. 5. Typical runs of tabu search with different neighbor selection strategies.

Table 1 compares the quality of the solutions obtained by tabu search and simulated annealing with different neighbor selection strategies. The algorithms have been run for five times with 30 minutes of CPU time given for each run. Further improvement has hardly been observed after 30 minutes. The numbers in the table are five time average of the evaluation values obtained by calculating the equation (1). We can see that NSPF with nonlinear bias performs the best compared to other methods. Improvement by NSPF with or without bias has been found consistent in both tabu search and simulated annealing search.

Table 1. Performances of tabu search and simulated annealing with different neighbor selection strategies.

Neighbor Selection Methods	Tabu Search	Simulated Annealing
Greedy	2.52	2.54
Random	1.82	1.79
NSPF (no bias)	1.65	1.69
NSPF ($\rho = 0.5$)	1.45	1.61

Table 2 shows for a few representative cases the total number of iterations, the average number (shown in parentheses) of candidate solutions generated for investigation during probabilistic filtering at each iteration, and the total number of solutions evaluated during the whole run of 30 minutes. The total number of iterations by NSPF is the smallest because of the overhead of neighbor selection by probabilistic filtering. However, NSPF shows the best performance because it can examine quite a large number of candidate solutions during probabilistic filtering. Simulated annealing has the largest total number of iterations but has the smallest number of full evaluations done because only one solution is evaluated at each iteration.

Table 2. The number of iterations, the number of solutions generated for investigation during probabilistic filtering, and the number of solutions evaluated for a few representative cases.

	Tabu Search				Simulated Annealing
	Random	Greedy	NSPF(no bias)	NSPF($\rho = 0.5$)	NSPF($\rho = 0.5$)
# iterations	65×10^3	40×10^3	37×10^3	34×10^3	730×10^3
# generated (per iteration)	1.50×10^6 (23)	0.92×10^6 (23)	14.7×10^6 (392)	15.4×10^6 (450)	14.9×10^6 (20)
# evaluated	1.50×10^6	0.92×10^6	0.86×10^6	0.78×10^6	0.73×10^6

5 Conclusions

We have proposed a neighbor selection method that can be used by local search algorithms so that they can investigate only a limited number of promising neighbors. The method works as a preprocessing in which candidate neighbors are randomly generated and evaluated by a simple heuristic measure and then only good-looking ones are selected based on a biased probabilistic decision. The local search algorithms employing this neighbor selection method enjoys the effect of virtually examining a large number of neighbors without fully evaluating all of them by the real objective function which is expensive to compute.

The bias function introduced for probabilistic decision making provides a convenient means of controlling the amount of bias to give towards the result of preliminary heuristic evaluation. Too much or less commitment (or bias) to the heuristic evaluation leads to either greedy or random search, respectively. What is important is to maintain a good balance between exploration and exploitation. Experiments with a large-scaled load balancing problem have shown that our selection method with biased probabilistic filtering outperforms both greedy and random selection methods.

References

1. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics 34 (1984) 975-986
2. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers (1997)
3. Selman, B., Kautz, H.A.: Domain-Independent Extensions to GSAT: Solving Large Structured Satisfiability Problems. Proceedings of IJCAI-93, Chambery, France (1993) 290-295
4. Selman, B., Levesque, H.J., Mitchell, D.G.: A New Method for Solving Hard Satisfiability Problems. Proceedings of AAAI-92, San Jose, CA (1992) 440-446
5. Bresina, J.L.: Heuristic-Biased Stochastic Sampling. Proceedings of AAAI-96 (1996) 271-278
6. Langley, P.: Systematic and Non-Systematic Search. Proceedings of AIPS-92, College Park, MD. Morgan Kaufmann Publishers (1992)
7. Minton, S., Bresina, J., Drummond, M.: Total Order and Partial-Order Planning: A Comparative Analysis. Journal of AI Research 2. AI Access Foundation & Morgan Kaufmann Publishers (1994) 227-262
8. Binato, S., Hery, W.J., Loewenstern, D., Resende, M.G.C.: A GRASP for job shop scheduling. In: Ribeiro, C.C., Hansen, P. (eds.): Essays and surveys on metaheuristics. Kluwer Academic Publishers (2001) 59-79
9. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6 (1995) 109-133

Systematic versus Non Systematic Methods for Solving Incremental Satisfiability

Malek Mouhoub and Samira Sadaoui

Department of Computer Science
University of Regina
3737 Waskana Parkway,
Regina SK, Canada, S4S 0A2
{mouhoubm,sadaouis}@cs.uregina.ca

Abstract. Propositional satisfiability (SAT) problem is fundamental to the theory of NP-completeness. Indeed, using the concept of "polynomial-time reducibility" all NP-complete problems can be polynomially reduced to SAT. Thus, any new technique for satisfiability problems will lead to general approaches for thousands of hard combinatorial problems. In this paper, we introduce the incremental propositional satisfiability problem that consists of maintaining the satisfiability of a propositional formula anytime a conjunction of new clauses is added. More precisely, the goal here is to check whether a solution to a SAT problem continues to be a solution anytime a new set of clauses is added and if not, whether the solution can be modified efficiently to satisfy the old formula and the new clauses. We will study the applicability of systematic and approximation methods for solving incremental SAT problems. The systematic method is based on the branch and bound technique while the approximation methods rely on stochastic local search and genetic algorithms.

Keywords: Propositional Satisfiability, Local Search, Genetic Algorithms, Branch and Bound.

1 Introduction

A boolean variable is a variable that can have one of two values: *true* or *false*. If x is a boolean variable, ¬ x is the negation of x. That is, x is true if and only if ¬x is false. A *literal* is a boolean variable or its negation. A *clause* is a sequence of literals separated by the logical *or* operator (∨). A logical expression in *conjunctive normal form (CNF)* is a sequence of clauses separated by the logical *and* operator (∧). For example, the following is a logical expression in CNF:

$$(x_1 \vee x_3) \wedge (\neg x_1 \vee x_2) \wedge \neg x_3 \qquad (1)$$

The *CNF-Satisfiability Decision Problem* (called also SAT problem) is to determine, for a given logical expression in CNF, whether there is some truth assignment (set of assignments of true and false to the boolean variables) that makes the expression true. For example, the answer is "yes" for the above CNF expression since the truth assignment $\{x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}\}$ makes the expression true. SAT problem is fundamental to the theory of NP-completeness. Indeed, using the concept

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 543-551, 2004.

of "polynomial-time reducibility" all NP-complete problems can be polynomially reduced to SAT[1]. This means that any new technique for SAT problems will lead to general approaches for thousands of hard combinatorial problems.

One important issue when dealing with SAT problems is to be able to maintain the satisfiability of a propositional formula anytime a conjunction of new clauses is added. That is to check whether a solution to a SAT problem continues to be a solution anytime a set of new clauses is added and if not, whether the solution can be modified efficiently to satisfy the old formula and the new clauses.

In this paper we will investigate different systematic and approximation methods for solving the SAT problem in an incremental way. The systematic method is a branch and bound technique based on the Davis-Putnam-Loveland algorithm. The second method relies on stochastic local search. Indeed the underlying local search paradigm is well suited for recovering solutions after local changes (addition of constraints) of the problem occur. The third method, based on genetic algorithms, is similar to the second one except that the search is multi-directional and maintains a list of potential solutions (population of individuals) instead of a single one. This has the advantage to allow the competition between solutions of the same population which simulates the natural process of evolution. Experimental comparison of the different methods on randomly generated SAT instances favors the approximation methods (stochastic local search and genetic algorithms) over the systematic one (branch and bound). The approximation methods however do not guarantee the correctness of the solution provided.

Note that related work on solving SAT problems in an incremental way has already been reported in the literature. These methods however rely solely on stochastic local search [1,2] or systematic search (backtrack search or branch and bound) [3,4,5] while our goal is to explore and compare different systematic and approximation methods to tackle the dynamic satisfiability problem.

In the next section we define the dynamic satisfiability problem and present the corresponding resolution procedure. Sections 3, 4 and 5 are respectively dedicated to the systematic method based on branch and bound, the approximation method based on stochastic local search and the approximation method based on genetic algorithms. Section 6 is dedicated to the empirical experimentation evaluating the three methods. Concluding remarks and possible perspectives are finally presented in conclusion.

2 Solving Dynamic SAT Problems

Before we present the general procedure for solving incremental SAT formulas, let us define the dynamic SAT problem.

2.1 The Dynamic CNF-Satisfiability (DSAT) Problem

We define a dynamic SAT problem (DSAT) as a sequence of static SAT problems $SAT_0, \ldots, SAT_i, SAT_{i+1}, \ldots, SAT_n$ each resulting from a change in the preceding one

[1] We will refer the reader to the paper published by Cook [6] proving that if CNF-Satisfiability is in P, then P = NP.

imposed by the "outside world". This change can either be a restriction (adding a new set of clauses) or a relaxation (removing a set of clauses because these later clauses are no longer interesting or because the current SAT has no solution). In this paper we will focus only on restrictions. More precisely, SAT_{i+1} is obtained by performing an addition of a set of clauses to SAT_i. We consider that SAT_0 (initial SAT) has an empty set of clauses. A DSAT over a set X of boolean variables is a sequence of static SAT problems using only the variables in X. Solving a DSAT problem consists of maintaining the satisfiability of the related static SAT problems anytime a new set of clauses is added.

2.2 General Procedure for Solving DSAT

Let us assume that we have the following situation: $SAT_{i+1} = SAT_i \wedge NC$ where:

- SAT_i is the current SAT formula,
- NC is a new set of clauses to be added to SAT_i,
- and SAT_{i+1} is the new formula obtained after adding the new set of clauses.

Both SAT_i and NC (and by consequence SAT_{i+1}) are defined on a set X of boolean variables. Asuming that SAT_i is satisfiable, the goal here is to check the consistency of SAT_{i+1} when adding the new set of clauses denoted by NC. To do so, we have defined the following procedure:

1. If $x \wedge \neg x$ is contained in NC, return that NC is inconsistent. NC cannot be added to SAT_i .
2. Simplify NC by removing any clause containing a disjunction of the form $x \vee \neg x$.
3. Let $NC = NC_1 \wedge NC_2$ where NC_1 is the set of clauses, each containing at least one variable that appears in SAT_i and NC_2 the set of clauses that do not contain any variable that appears in SAT_i or NC_1. Let $SAT_i = S_1 \wedge S_2$ where S_1 is the set of clauses, each containing at least one variable that appears in NC and S_2 the set of clauses that do not contain any variable that appears in NC or S_1. S_2 will be discarded from the rest of the procedure since any assignment to the variables of NC will not affect the truth assignment already obtained for S_2.
4. Assign the truth assignment of SAT_i to NC_1. If NC_1 is satisfiable goto 7.
5. Using a search method flip the variables of NC_1 that do not appear in S_1. If NC_1 is satisfied goto 7.
6. Using a search method, look for a truth assignment for both S_1 and NC_1. If no such assignment is found return NC cannot be added as it will affect the consistency of SAT_i.
7. Using a search method look for a truth assignment for NC_2. If no such assignment is found return NC cannot be added as it will affect the satisfiability of SAT_i.

Step 5 requires a search procedure that starts from an initial configuration and iterates until a truth assignment satisfying NC_1 is found. To perform this step we can use one of the following methods:

- A randomized local search method starting from the initial configuration. The local search algorithm will iterate by flipping the values of the variables of NC_1 that do not appear in S_1 until a truth assignment for NC_1 is found. Details about the stochastic local search method are presented in section 4.

- A genetic algorithm starting from a population containing instances of the initial configuration. The genetic algorithm iterates performing the mutation and crossover operators on only the part of the vectors containing the variables of NC_1 that do not appear in S_1. The method based on genetic algorithms is presented in section 5.
- A branch and bound method which starts with a lower bound equal to the number of non satisfied clauses of the initial configuration, and explores a subset of the search space by assigning values to the variables that appear in NC_1 and not S_1. The algorithm will stop when the lower bound is equal to zero or when the entire subset of the search space is explored. The detail of the branch and bound method is presented in section 3.

In step 6 the search procedure starts from the best configuration (assignment) found in step 5, and iterates until a truth assignment satisfying both NC_1 and S_1 is obtained. The search procedure has also to make sure to avoid checking any configuration already explored in step 5. This can be done by checking, at each variable assignment, that the subset of variables belonging to S_1 and that do not belong to NC_1 has an assignment different from the old one satisfying SAT_i. Step 7 requires a search procedure for determining a truth assignment for NC_2.

3 Solving SAT Using Systematic Search Techniques

The exact algorithms for solving the satisfiability problem include the well known Davis-Putnam-Loveland algorithm [7,8] and the integer programming approaches [9]. As we have seen in the previous section, we are mainly concerned with a procedure that solves the SAT problem by increasing at each step the number of satisfied formulas. Thus, the algorithm that we will use is a branch and bound variant of the Davis-Putnam-Loveland procedure. This algorithm starts with an upper bound (*UB*) corresponding to the number of unsatisfied clauses of a given complete assignment. The algorithm will then iterate updating the value of *UB* anytime a new complete assignment with a lower number of unsatisfied clauses is found. The algorithm will stop when *UB* is equal to zero which corresponds to a solution satisfying all the clauses. Our implementation of the algorithm is as follows:

We use a variant of the backtrack search method that compares at each node the upper bound UB with a lower bound LB corresponding to the sum of the number of unsatisfied clauses of the current partial assignment and an underestimation of the number of clauses that become unsatisfied if we extend the current partial assignment into a complete one. If $UB \leq LB$ the algorithm backtracks and changes the decision at the upper level. If $UB > LB$ the current partial assignment is extended by instantiating the current node to true or false. If the current node is a leaf node, UB will take the value of LB (a new upper bound has been found). The algorithm will stop when UB is equal to zero. The underestimation is equal here to the minimum between the number of clauses that become unsatisfied if true is chosen for the next assignment and the number of clauses that become unsatisfied if false is chosen for the next assignment. For choosing the next variable to assign, we use the in-most-shortest clause heuristic as reported in [10].

4 Solving SAT Using Stochastic Local Search

One of the well known randomized local search algorithms for solving SAT problems is the GSAT procedure [11,12] presented below. GSAT is a greedy based algorithm that starts with a random assignment of values to boolean variables. It then iterates by selecting at each step a variable, flips its value from *false* to *true* or *true* to *false* and records the decrease in the number of unsatisfied clauses. The algorithm stops and returns a solution if the number of unsatisfied clauses is equal to zero. After MAX-FLIPS iterations, the algorithm updates the current solution to the new solution that has the largest decrease in unsatisfied clauses and starts flipping again until a solution satisfying all the clauses is found or MAX-TRIES is reached. This is how the algorithm is used in step 7 of our general procedure. When used in step 5, the GSAT algorithm starts from the initial configuration (corresponding to the truth assignment found for the formula before adding the new clauses) instead of a random configuration. Also, only the variables belonging to NC_1 and which do not appear in S_1 can be chosen for the flip. For step 6, GSAT starts from the best configuration found in step 5. Also, all the configurations explored in step 5 are avoided in step 6 as shown in subsection 2.2.

```
Procedure GSAT
    begin
       for  i ← 1  until MAX-TRIES do
       begin
          T ← a randomly generated truth assignment
          for  j ← 1  until MAX-FLIPS do
             if  T  satisfies the formula then
                   return  T
             else make a flip
       end
       return ("no satisfying assignment found")
    end
```

5 Solving SAT Using Genetic Algorithms

Genetic algorithms [13] (GAs) perform multi-directional non systematic searches by maintaining a population of individuals (called also potential solutions) and encouraging information formation and exchange between these directions. It is an iterative procedure that maintains a constant size population of candidate solutions. Each iteration is called a generation and it undergoes some changes. *Crossover* and *mutation* are the two primary genetic operators that generate or exchange information in GAs. Under each generation, *good solutions* are expected to be produced and *bad solutions* die. It is the role of the objective (evaluation or fitness) function to distinguish the goodness of the solution. The idea of crossover operators is to combine the information from parents and to produce a child that obtains the characteristics of its ancestors. In contrast, mutation is a unary operator that needs only one input. During the process, mutation operators produce a child by selecting some bad genes from the parent and replacing them with the good genes. The two operators may behave differently but they both follow the characteristic of GAs in that the next generation is expected to perform better than the ancestors.

Let us see now how to solve the CNF-Satisfiability problem using genetic algorithms. Since we are dealing with variables that can only take on two states, true or false, the representation we choose is a binary vector of length n where n is the number of boolean variables with the coding of 1 being true and 0 being false. Each entry in the vector corresponds to the truth assignment for the variable which corresponds to that location in the vector. For example, if we consider the formula in introduction, then the vector (010) will correspond to the truth assignment {x_1 = false , x_2 = true , x_3 = false} which does not satisfy the formula.

The pseudo code of the GA based search algorithm we use is presented below.

```
begin
  t ← 1
  // P(t) denotes a population at iteration t
  P(t) ←  n randomly generated individuals
  eval←  evaluate P(t)
  while termination condition not satisfied do
    begin
      t ← t + 1
      select  P(t)  from  P(t-1)
      alter  P(t)
      evaluate  P(t)
    end
end
```

The GA search method starts from an initial randomized population of individuals and evaluates each vector using a fitness function to see if we have discovered the optimum solution. We define the fitness function as the number of true clauses corresponding to a given vector. For example, the fitness function of the vector (110) is equal to 2. If the fitness function is equal to C (C is the number of clauses of the formula) then the CNF expression is satisfied. After evaluating the randomized population, if the optimum function is not found then the crossover and mutation operators will be applied to some selected individuals. The way we use to select the individuals (*select* function) is to assign a probability of being selected to each individual in proportion of their relative fitness. That is, an individual with the fitness function equal to 10 is 10 times more likely to be chosen than an individual with a score of 1. Note that we may obtain multiple copies of individuals that happened to be chosen more than once (case, for example, of individuals with good fitness function) and some individuals very likely would not be selected at all. Note also that even the individual with the best fitness function might not be selected, just by random chance. In step 7 of our resolution procedure the above GA method will be used as is to look for the satisfiability of the formula NC_2. In step 5, the initial population contains instances of the initial configuration. Crossover and mutation operators are modified such that only the entries of the vectors corresponding to variables of N_1 which do not appear in S_1 are affected by the operators. The crossover operator randomly selects two individuals (that we call parents) and generates a new individual (that we call child) as follows. We assume N the number of variables in the vector and NS a random number chosen from [1..N]. The child individual is randomly generated from the two parents by taking the first NS variables from the first parent and the last N-NS variables from the second parent. The mutation operator selects one individual and flips the value of one of its variables chosen at random.

6 Experimentation

In this section we will present an experimental comparison of the following four methods for solving DSAT problems.

- **SLS**: the stochastic local search method is used here in steps 5,6 and 7 of the resolution procedure.
- **GA**: the method based on genetic algorithms is used in steps 5,6 and 7.
- **BB**: the branch and bound method is used in steps 5, 6 and 7.
- **BB+SLS**: the branch and bound method is used in steps 5 and 6 while the stochastic local search method is used in step 7.

The experimental tests are performed on randomly generated DSAT instances. Since we did not found libraries providing DSAT problems, we randomly created these instances as follows:

1. First, random 3-SAT instances are taken from the well known SATLIB library (`www.informatik.tu-darmstadt.de/AI/SATLIB`). Each instance, characterized by the number of variables n and clauses k it contains, is generated as follows. Each of the k clauses is constructed from 3 literals which are randomly chosen from the 2n possible literals (the n variables and their negations) such that each possible literal is selected with the same probability of 1/2n. Clauses are not accepted for the construction of the problem instance if they contain multiple copies of the same literal or if they are tautological (i.e., they contain a variable and its negation as a literal). All random 3-SAT instances are satisfiable. More details about this generation process can be found in [14].
2. Each DSAT instance is then developed from a 3-SAT one in a series of stages. At each stage, a random number of clauses is taken from the 3-SAT instance and added to the DSAT one until there are no more clauses to take. The initial DSAT has 0 clauses. In our experimentation, the number of stages is fixed to 10. Let us assume N the total number of clauses of the 3-SAT instance and N_1, N_2, … , N_{10} the random numbers of clauses at each stage. These numbers are generated as follows. N_1 and N_2 are randomly chosen from [1,N/5 – 1]. N_3 and N_4 will then be generated from [1, (N – N_1 – N_2)/4 – 1], and N_5 and N_6 chosen from [1, (N – N_1 – N_2 – N_3 – N_4)/3 – 1]. N_7, N_8, N_9 and N_{10} will be generated in the same manner. This will guarantee that the numbers N_1, N_2, … , N_{10} will each have a value almost equal to N/10.

Table 1 presents the time performance in seconds needed by each method to maintain the satisfiability of randomly generated DSAT instances. Indeed, each method is first executed on a given number of instances (100 or 1000) from each test-set. We take then the average running time required to achieve the satisfiability of the instances. All tests are performed on a 2GHz Pentium IV computer under Linux. The “- ” symbol indicates that the corresponding method fails to obtain the satisfiability of instances for a given test-set.

SLS and GA approximation methods present the best results. The performances of these two methods are comparable. Further study of the search space is needed in order to see when to expect genetic algorithms to outperform stochastic local search and vice versa. Due to the exponential running time of the branch and bound method (comparing to the polynomial time cost of the approximation methods) BB and BB+SLS are slower especially for large instances.

Table 1. Performance of the four methods on randomly generated DSAT problems

Test Set	# of instances	# of variables	# of clauses	SLS	GA	BB	BB+SLS
uf20-91	**1000**	**20**	**91**	**0.001**	**0.01**	**0.2**	**0.2**
uf50-218	**1000**	**50**	**218**	**0.004**	**0.02**	**1.2**	**0.8**
uf75-325	**100**	**75**	**325**	**0.01**	**0.06**	**1.6**	**0.9**
uf100-430	**1000**	**100**	**430**	**0.08**	**0.2**	**2.4**	**1.4**
uf125-538	**100**	**125**	**538**	**0.132**	**0.6**	**4.4**	**2.8**
uf150-645	**100**	**150**	**645**	**0.227**	**1.2**	**12.8**	**7.2**
uf175-753	**100**	**175**	**753**	**0.951**	**3.2**	**35.6**	**15.7**
uf200-860	**100**	**200**	**860**	**3.197**	**3.5**	**128**	**37**
uf225-960	**100**	**225**	**960**	**32.1**	**12.8**	**-**	**158**
uf250-1065	**100**	**250**	**1065**	**37.82**	**34**	**-**	**-**

7 Conclusion

In this paper we have presented different ways based respectively on systematic and approximation methods for maintaining the satisfiability of CNF propositional formulas in an incremental way. Our work is of interest to a large variety of applications that need to be processed in an evolutive environment. This can be the case of applications such as reactive scheduling and planning, dynamic combinatorial optimization, dynamic constraint satisfaction and machine learning in a dynamic environment.

One perspective of our work is to deal with retraction of clauses in an efficient way. Assume that during the search, a given clause (or a set of clauses) is removed. Would it be worthwhile to reconsider any decision made because of these clause(s) or would it be more costly than just continuing on with search. Another idea we will investigate in order to improve the performance of our general procedure consists of processing steps 4 until 6 and step 7 of our procedure in parallel. If any of these two parallel phases fails then the main procedure will stop and returns NC (set of new clauses to be added) inconsistent.

References

1. H.H. Hoos and K. O'Neil. Stochastic Local Search Methods for Dynamic SAT - an Initial Investigation. In AAAI-2000 Workshop on Leveraging Probability and Uncertainty in Computation, pages 22-26, 2000.
2. J. Gutierrez and A.D. Mali. Local Search for Incremental Satisfiability, In International Conference on Artificial Intelligence, pages 986-991, 2002.
3. J.N. Hooker. Solving the Incremental Satisfiability Problem. Journal of Logic Programming, vol. 15, pages 177-186, 1993.
4. H. Bennaceur, I. Gouachi and G. Plateau. An incremental Branch-and-Bound Method for Satisfiability Problem. INFORMS Journal on Computing, vol. 10, pages 301-308, 1998.
5. Jesse Whittemore, Joonyoung Kim and Karem A. Sakallah. SATIRE: A New Incremental Satisfiability Engine. DAC 2001, pages 542-545, 2001.
6. S.A. Cook. The complexity of theorem proving procedures. In 3rd Annual ACM Symposium on the Theory of Computing, pages 151-158, 1971.
7. M. Davis and H. Putnam. Journal of The Association for Computing Machinery, 7:201-215, 1960.
8. D. Loveland. Automated Theorem Proving: A Logical Basis. North Holland, 1978.
9. R.G. Jeroslow, C.E. Blair and J.K Lowe. Some results and experiments in programming techniques for propositional logic. Computers and Operations Research, 13(5):633-645, 1986.
10. R.J. Wallace and E.C. Freuder. Comparing constraint satisfaction and davis-putnam algorithms for the maximal satisfiability problem. In D.S. Johnson and M.A. Trick editors, Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge. American Mathematical Society, 1995.
11. B. Selman and H.A. Kautz. An empirical study of greedy local search for satisfiability testing. In AAAI'93, pages 46-51, 1993.
12. B. Selman, H.A. Kautz and B. Cohen. Noise Strategies for Improving Local Search. In AAAI'94, pages 337-343. MIT Press, 1994.
13. Z. Michalewicz. Genetic Algorithms + Data Structures = Evaluation Program. Springer-Verlag, 1992.
14. P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In IJCAI-91, pages 331-337, 1991.

Improved GRASP with Tabu Search for Vehicle Routing with Both Time Window and Limited Number of Vehicles

Zhiye Li[1], Songshan Guo[1], Fan Wang[2], Andrew Lim[2]

[1] Dept. of Computer Science, Zhongshan (Sun Yat-sen) University,
Guangzhou, 510275, China
[2] Dept. of Industrial Engineering and Engineering Management,
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Abstract. In this paper, we study a new useful extension of vehicle routing problem (VRP) – VRP with both time window and limited number of vehicles (m-VRPTW). We propose an improved Greedy Randomized Adaptive Search Procedure (GRASP) framework by techniques of multiple initializations, solution reuse and mutation improvement, with four specified heuristics for m-VRPTW: short left time first, near customer first, short waiting time first and long route first. From the experimental results on benchmark data, it is shown that our algorithm not only well solves the m-VRPTW problem, but also obtains accurate solution for classical VRPTW problem with stable performance in short running time. In fact, the search techniques proposed in this paper can be easily applied for other meta-heuristics for problem solving …

1 Introduction

The Vehicle Routing Problem (VRP) calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers, and it is one of the most important and studied combinatorial optimization problems for past 40 years. The VRP is a well-known integer-programming problem with falls into the category of NP-Complete problems. For such problems it is often desirable to obtain approximate solutions, provided they can be found fast enough and are sufficiently accurate for the purpose. The difficulty of solving VRP lies at the intersection of the following two well-studied problems: The Traveling Salesman Problem (TSP) and Bin Packing Problem (BPP). Because of the interplay between the two underlying problems, instances of the VRP can be extremely difficult to solve in practice. In real practice, VRP appears many constraints, such as capacitated VRP(CVRP), VRP with time window(VRPTW), VRP by multiple depots (MDVRP), VRP with pick-up and delivering (VRPPD) and periodic VRP(PVRP). These extensions of VRP are widely used in a lot of real-life applications including currency delivery and scheduling at ATM machines, dynamic sourcing and transport of fuels, pickup of charitable donations from homes, snow plow and snow removal routing, postal delivery truck routing and so on.

Most well-studied extensions of VRP focus on how to minimize the resources (such as the number of vehicles) to satisfy all customers under all constraints. In fact, in real practice, there exists another type of application of VRP that for given limit

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 552-561, 2004.

resources, how to satisfy as most requests as possible. For instance, if you are a boss of a transportation company, for a lot of incoming business orders, you must have decision on how many customer orders your company can serve at most by current limit resources such as vehicles, man-powers and so on. In addition, you may need to consider how to support high quality service to elite customers, how to support a robust planning of vehicle transportation to minimize the risk of uncertain events when you serve customers. In above cases, we not only consider the optimization of VRP, but also consider other factors of the solution, i.e. stability, robustness and risk.

In this paper, we study a case of above types of VRP, named Vehicle Routing Problem with Both Time Window and Limited Number of Vehicles (m-VRPTW). We propose an improved Greedy Randomized Adaptive Search Procedure (GRASP) framework by techniques of multiple initialization, solution reuse and mutation improvement. Four useful heuristics m-VRPTW are also presented, including short left time first, near customer first, short waiting time first and long route first. From the experimental results for well-studied benchmark test data, it is shown that our algorithm not only well solves the m-VRPTW problem studied in this paper, but also obtains good solution for classical VRPTW problem with stable performance in short running time. In fact, above search techniques proposed in this paper can be easily applied for other search frameworks in problem solving.

The paper is organized as follows. In Section 2, we state the m-VRPTW problem formally. In Section 3, we propose a new GRASP framework with many new techniques, such as multiple initial solutions, solution reuse and mutation improvement. Furthermore, we describe the complete algorithm for m-VRPTW in detail in Section 4 and give the computational results for both m-VRPTW and VRPTW for benchmark test data in Section 5. Lastly, we conclude the paper in Section 7.

2 Problem Statement

The m-VRPTW problem is defined as follows formally. There is an undirected graph $G(V, E)$ where $V=\{v_0, v_1, \ldots, v_n\}$ and $E=\{(v_i, v_j)\colon i \neq j,\ 1 \leq i, j \leq n\}$. v_0 represents the depot while v_i represents one customer with demand d_i, time window (e_i, l_i) and service duration s_i, for all $1 \leq i \leq n$. In addition, we assign a travel distance (time) t_{ij} to each edge (v_i, v_j). And then, we have total m vehicles with the capacity of C. The objective function of the m-VRPTW is to maximize the number of customers served by the m routes $\{R_1, R_2, \ldots, R_m\}$, one route by one vehicle, satisfying the following constraints.

1. Each route must starts and ends at depot v_0;
2. The total demands among all served customs in one route should not exceed the vehicle capacity, i.e. $\sum_{v_i \subseteq R_j} d_i \leq C, 1 \leq j \leq m$;
3. Each customer can only be served by one vehicle (in one route) within its corresponding time window. In other words, for each customer, it is either served by one vehicle or not served by any vehicle at all, e.g. $v_i \neq v_j$ if vi $\in R_p$ and $v_j \in R_q$ $(p \neq q)$.

From the above definition, we can know that m-VRPTW is a generalization of the multiple constrained knapsack problems which have been proved as NP-hard problems. Because of NP-hard, we should find approximation algorithms with high performance to make the computation in realistic-size reliable.

To our knowledge, there has little research that focus on m-VRPTW in literature. We firstly give brief literature review for the classical VRPTW problem. The VRPTW problem is NP-hard and even finding a feasible solution is also a NP-hard [11]. The early work on the VRPTW was case study oriented [6], [2]. Later research focused on the design of heuristics and the development of effective optimal approach. In heuristics, route construction algorithms build a feasible solution by inserting an un-routed customer into current partial route, such as sequential insertion heuristics by Solomon [9]. Route improvement methods iteratively modify the current solution by performing local searches for better neighboring solutions by *r*-exchange operator [15], [1], [10], [13]. Another type of heuristics is composite heuristics that blend route construction and improvement algorithms, for example, a combination of greedy heuristic and randomization to produce initial routes in parallel [2]; embedding route improvement within the tour construction process [16] and hierarchical local search [14]. On the other hand, many efficient meta-heuristics have been proposed for VRPTW problem including Rochat and Taillard [21], Potvin and Rousseau [5], Chiang and Russell [20]. Another alternative used in meta-heuristics is guided local search described by Kilby et al [12]. Moreover, Thangiah et al combined genetic algorithm, simulated annealing and tabu search for VRPTW [17]. Only Lau et al proposed an ordinary tabu search algorithm [4]. In that algorithm, a holding list is used to contain the list of customers that haven't been served in the current solution. The customers are then transferred back and forth the holding list under the tabu search strategy. The author combined the two-stage approach into a nested approach. He increased the number of vehicles in stages and at each stage, applies standard tabu search to maximize the number of customers to be inserted into vehicles. In general, the experimental results published in the paper leave great space for us to improve.

3 Improved GRASP

The Greedy Randomized Adaptive Search Procedure (GRASP) is a two-phase multi-start method that consists of a construction phase in which a feasible solution is constructed iteratively followed by a local search phrase that tries to improve each constructed solution. GRASP was first applied to a difficult set covering problem [18]. [7] provided a recent survey of the basic method, enhancements and applications of GRASP.

We illustrate the classical GRASP as follows. In the first phase of the GRASP, a feasible solution is constructed by greedy randomization. We call this step is greedy randomized construction. In each iteration of this phase, let the set of candidate elements be formed by all elements that can be incorporated to the partial solution under construction without destroying feasibility. The selection of the next element for incorporation is determined by the evaluation of all candidate elements according to a greedy evaluation function. The second phase of the GRASP is a local search to find a local optimal solution starting from the solution given by the greedy randomized construction. If the current local optimal solution has better performance than the stored best solution, we update the best solution. These two phases will be iterated several times.

As pointed out by [2], for GRASP to work well, it is important that high-quality solutions (feasible solutions that are close to the optimum as tight as possible) are constructed during the first phase, with the most promising solutions passed to the second phase. This is different from Tabu search and simulated annealing which do not need high-quality feasible initial solutions and expend much effort in trying to improve the current solution. From this motivation, we propose some improvements on classical GRASP to obtain better performance under limit time consuming, by the following three basic ideas:

1. Create multiple initial solutions by heuristics at greedy or at random. we call it "multiple initial solutions";
2. Inherit the local optimum found by the local search as the starting search solution for greedy randomized construction in next iteration. We call this strategy "Solution reuse";
3. During the iterative search procedure in GRASP, we randomly mutate current solution to jump the current search subspace for performance improvement if possible. Moreover, the occurrence of the mutation is controlled by a given probability. We call this technique "Mutation Improvement".

3.1 Multiple Initial Solutions

For m-VRPTW, we create the following six different methods to construct initial solutions for GRASP. The first four heuristics are based on various greedy criteria.

1. Short left time first: try to insert every unserved customer into the end of each route to minimize the left time.
2. Near customer first: construct the route for each vehicle one by one. For each route, insert the unserved customer that provides the shortest additional left time at the end of the route. The above insertion will be iterated until there is no unserved customer can be inserted into that route.
3. Short waiting time first: construct the route for each vehicle one by one. For each route, insert the unserved customer at the most suitable position that provides the shortest total waiting time due to the time windows constraints of the customers. The above insertion will be iterated until there is no unserved customer can be inserted into that route.
4. Long route first: construct the route for each vehicle one by one. For each route, insert the unserved customer at the most suitable position that makes that route has the longest length expansion. The above insertion will be iterated until there is no unserved customer can be inserted into that route.

Moreover, we also develop two random methods to get random initial solutions.

1. Random customer insertion: select unserved customer one by one randomly. For each selected customer, randomly select one route that the customer can be inserted into, if there is such a route. And then, insert the customer into that route at the position where the left time of that route is minimized.
2. Random route construction: select an uncompleted route randomly to construct. For that route, select randomly an unserved customer to be inserted into it, if there is such a customer can be inserted into it. Otherwise, block this route as complete constructed. The above procedure will be iterated many times until all routes have been blocked.

3.2 Solution Reuse

The basic GRASP produces a new initial solution in every iteration without any relationship with the results from the local search of the previous iterations. That means GRASP never reuses the previous results in history during search. In our algorithm, we try to modify the GRASP framework to reuse the solution from the previous iteration as the input for greedy randomized construction in current iteration.

Different from the classical GRASP, in the modified algorithm we reuse the result from the local search as the input for the greedy randomized construction of the next iteration. To make the idea of solution reuse reliable, we must modify the original greedy randomized construction procedure to allow the construction starting from a given solution. Such a modification may have a lot of variations depending on the problems. In our algorithm to solve the m-VRPTW, we use the random reassignment to implement it. For each route r_i ($1 \le i \le m$) in a given solution, randomly select a pair of two customers in that route and try to exchange their positions. If the new assignment is a feasible solution, update the exchange. Otherwise, keep the original solution. The above random reassignment will be applied to all m routes.

3.3 Mutation Improvement

From the experience of applying genetic algorithm or other meta-heuristics for solving difficult NP problems, the mutation always benefits the search procedure to get more accuracy result. In the second phrase of the GRASP, normally the determined local search without any random component works stably. However, such a stable search always loses the energy. In the other hand, a search algorithm with too many random components can help the search procedure jump the local optimum and bring the search to enter an untouched search subspace. However, the performance of random search is not stable: sometimes gets wonderful results but sometimes gets poor results. In our algorithm to solve m-VRPTW problem, we bring a random component in the framework of GRASP after local search. In addition, we also set one probability parameter to balance between the randomization and determination in search.

After the local search in each iteration of GRASP, we repeat the following random search by fixed Ta_{max} times for each route r_i ($1 \le i \le m$) of the solution. That is to randomly select a pair of two customers in the route r_i, and then try to exchange their positions in that route to get a new route. If the new route not only is feasible but also has shorter left time than the original one, update the exchange and exit the iteration of the random search for the current route r_i. The occurrence of this mutation is controlled by a fixed probability parameter p ($0 \le p \le 1$), that means it will be run p*100 times per 100 iterations in the GRASP.

4 Algorithm

The whole algorithm to solve m-VRPTW is illustrated as Algorithm 1. The algorithm first calls the six methods proposed in Section 3.1 to generate six initial solutions (four at greedy and two at random) for the following iterated search of GRASP. The global iterative search in GRASP will be continued until the iteration time once exceeds the fixed maximum iteration time - *MaxIteration*. For each initial solution from previous iteration (in the first iteration, use the results from the six algorithms for initial solution generation, we process them following the four steps:

1. Tabu search: call standard tabu search (iteration time is equal to *T_Tabu*) to find local optimum from the initial solution;
2. Mutation: Call random search proposed in Section 3.3 to improve the local optimum under the probability *p*;
3. Update: update the current optimal solution if a better solution has been found;
4. Solution reuse: call random generation procedure that mentioned in Section 3.2 to generate an initial solution for next iteration via current search result.

In the algorithm, let x^* be the best solution found; let f be the solution-elevated function, and $f^*=f(x^*)$.

Algorithm 1 Improved GRASP with smoothed dynamic tabu search for m-VRPTW

```
f* ← ∞;
Generate six initial solutions $x^1, x^2, \ldots, x^K$ ($K = 6$)
Iteration, Iteration' ← 0;
while Iteration < MaxIteration do
    Iteration ← Iteration + 1
    if Iteration' < Iteration then
        Iteration' ←Iteration' + 1
    else
        Iteration' ←Iteration' - 1
    end if
    $TabuTenure = \left\lfloor \frac{MaxTabuTenure \times (MaxIteration - Iteration')}{MaxIteration} \right\rfloor$
    for $1 \leq i \leq K$ do
        Call tabu search to find the local optimum $newx^i$ from $x^i$;
        Call random search for $newx^i$ under the probability $p$;
        if f(newx') < f* then;
            f* ← f(newx');
            x* ← newx';
            Iteration' ← 0;
        end if
        Call random assignment to get x from newx';
    end for
end while
return x*;
```

5 Experimental Results and Analysis

5.1 Test Data and Experimental Environment

To evaluate our improved GRASP algorithm with tabu search technique to solve the m-VRPTW problem, we tested its performance on the set of 56 Solomon's test cases with 100 customers [8]. The 56 test cases (six sets) are designed highlights several factors that affect the behavior of routing and scheduling algorithms including geographical data; the number of customers serviced by a vehicle; percent of time-constrained customers; and tightness and positioning of the time windows. The geographical data are randomly generated in sets R1 and R2, clustered in sets C1 and C2, and a mix of random and clustered structures in sets RC1 and RC2. Sets R1, C1 and RC1 have a short scheduling horizon and allow only a few customers per route (approximately 5 to 10). In contrast, the sets R2, C2 and RC2 have a long scheduling horizon permitting many customers (more than 30) to be serviced by the same vehicle. The customer coordinates are identical for all sets within one type.

We set the parameters of our algorithm as follows: *MaxIteration*=2000, *p*=0.2, *MaxTabuTenure*=30. The algorithm was tested on an Intel Celeron 4 Machine with 1.7G CPU and 512M RAM. The average running time of the algorithm for all test cases is around 2 minutes.

5.2 Performance on m-VRPTW

To access the performance of our proposed algorithm clearly, we illustrated the average number of customers served vs. different available vehicles of the upper bound and our GRASP in Figure 1, 2 and 3 for set C, R and RC. Moreover, since Lau didn't propose his results for neither set R nor set RC, we only could give the performance curve of his published m-VRPTW algorithm in the same condition for set C1 for comparison in Figure 1.

Anyway, we use ILOG CPLEX to computer all below upper bounds.

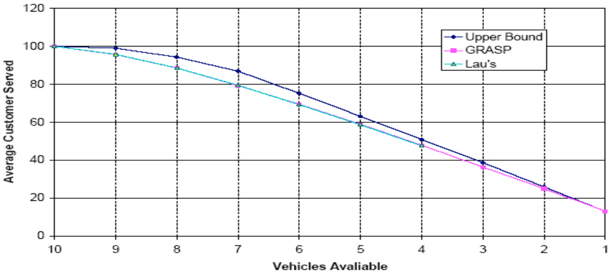


Fig.1. For set C1

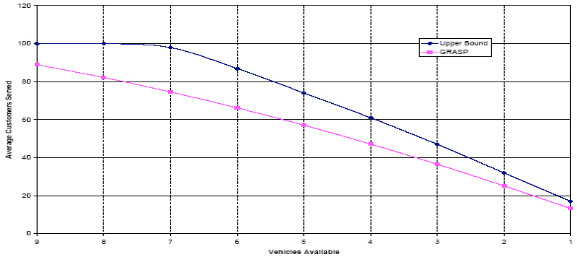


Fig.2. For set R1

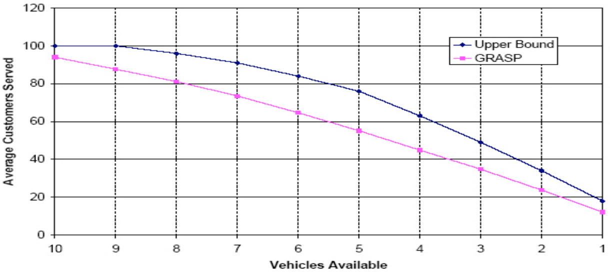


Fig.3. For set RC1

From the above given results, it is shown that

1. For set C1, the performance of our proposed algorithm is tight to the upper bound. Moreover, the performance of ours and Lau's are the same mostly.
2. For set R1 and RC1, our algorithm not only gives close results to the upper bound, but also shows stable performance when we increase or decrease the number of available vehicles. The gaps of the results between our algorithm and the upper bound are 0.68, 3.08 and 3.40 customers per vehicle. The gaps of set R1 and RC1 are much bigger than the one of set C1 is due to the random data distribution for R1 and RC1 rather than the clustering distribution for C1. In m-VRPTW, the binary programming model (upper bound) will leave more far away from the optimal solution for random distribution data than for clustering distribution data.

5.3 Performance on VRPTW

It is obviously that we can easily obtain the results for the classical VRPTW problem by the algorithms for m-VRPTW directly, by repeating the m-VRPTW algorithm while increasing the number of vehicles step by step until all the customers can be served completely. The result of m-VRPTW at that time (all customers can be served) is of course the result of VRPTW.

We support a summary of the average performance comparison among most current overall best published heuristics, Lau's heuristics for m-VRPTW and our improved GRASP for m-VRPTW, shown in Table 1. Here, the current best published heuristics include Rochat and Taillard (RT), Chiang and Russell (CR), Taillard et al. (TBGGP), Homberger and Gehring (HG) and Cordeau et al. (CLM).

By the above results and comparison, we have

1. Our proposed algorithm can solve the classical VRPTW problem well with the close performance to the best published results, even the objective function of our algorithm is not for VRPTW problem directly. Especially, for set C1, C2 and R2, our proposed algorithm achieves the best solutions in term of the minimizing the number of vehicles.
2. Our algorithm outperforms Lau's published heuristic in average performance for all sets, especially for set R and RC. In all 56 Solomon's test data, our improved GRASP with smoothed dynamic tabu search improves Lau's for 12 cases (21% of all cases) totally.

Table 1. The average results comparison

	RT	CR	TBGGP	HG	CLM	Lau	GRASP
C1 mean	10	10	10	10	10	10	10
C2 mean	3	3	3	3	3	3	3
R1 mean	12.83	12.17	12.17	11.92	12.08	12.92	12.42
R2 mean	3.18	2.73	2.82	2.73	2.73	3	2.73
RC1 mean	12.75	11.88	11.5	11.5	11.5	12.25	11.75
RC2 mean	3.65	3.25	3.38	3.25	3.25	3.38	3.375

6 Conclusion

In this paper, we study the vehicle routing problem with constraints of both time window and limit number of vehicle. This variant of VRPTW has wider applications in real business world. We improve GRASP framework by techniques of multiple initialization, solution reuse and mutation improvement. From the experimental results for well-studied benchmark test data, it has shown that our proposed algorithm not only solved the m-VRPTW problem studied in this paper, but also obtained accurate solution for the classical VRPTW problem with stable performance in short running time. In fact, above proposed search techniques can be easily applied to other meta-heuristics for problem solving.

References

1. Baker E. and Schaffer J. Computational experience with branch exchange heuristics for vehicle routing problems with time window constraints. American Journal of Mathematical and Management Sciences, 6:261–300, 1986.
2. Kontoravdis G. and Bard J.F. A grasp for the vehicle routing problem with time windows. ORSA Journal on Computing, 7:10–23, 1995.
3. Pullen H. and Webb M. A computer application to a transport scheduling problem. Computer Journal, 10:10–13, 1967.
4. Lau H.C., Sim Melvyn, and Teo K.M. Vehicle routing problem with time windows and a limited number of vehicles. European Journal of Operational Research, 148(3):559–569, 2003.

5. Potvin J.-Y. and Rousseau J.-M. An exchange heuristic for routing problems with time windows. Journal of Operational Research Society, 46:1433–1446, 1995.
6. Knight K. and Hofer J. Vehicle scheduling with timed and connected calls: A case study. Operational Research Quarterly, 19:299–310, 1968.
7. Resend M. Greedy randomized adaptive search procedures (grasp), 1998. Technical Report 98.41.1. AT&T Labs Research, Florham Park, NJ.
8. Solomon M.M. http://web.cba.neu.edu/ msolomon/problems.htm.
9. Solomon M.M. Algorithms for vehicle routing and scheduling problems with time window constraints. Operations Research, 32:254–265, 1987.
10. Solomon M.M., Barker E., and Schaffer J. Vehicle routing and scheduling problems with time windows constraints: Efficient implementations of solution improvement procedures. In Golden B.L and Assad A.A., editors, Vehicle routing: Methods and studies, pages 85–106. North-Holland, Amsterdam, 1988.
11. Savelsbergh M.W.P. An efficient implementation of local search algorithms for constrained routing problems. Annals of Operations Research, 4:285–305, 1985.
12. Kilby P.J., Prosser P., and Shaw P. Guided local search for the vehicle routing problem with time windows. In Voss S., Martello S., Osman I.H., and Roucairol C., editors, Meta Heuristics: Advances and Trends in Local Search Paradigms for Optimisation, pages 473–486. Kluwer, Boston, MA, 1998.
13. Thompson P.M. and Psaraftis H.N. Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. Operations Research, 41:935–946, 1993.
14. Cordone R. and Wolfer Calvo R. Note on time window constraints in routing problems, 1996. Internal Report 96.005, Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milan, Italy.
15. Russell R.A. An effective heuristics for the m-tour traveling salesman problem with some side conditions. Operations Research, 25:517–524, 1977.
16. Russell R.A. Hybrid heuristics for the vehicle routing problem with time windows. Transportation Science, 29:156–166, 1995.
17. Thangiah S.R., Osman I.H., and Sun T. Hybird genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows, 1994. Technical Report UKC/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK.
18. Feo T. and Resende M. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research letters, 8(2):67–71, 1989.
19. Chiang W.-C. and Russell R.A. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. INFORMS Journal on Computing, 9:417–430, 1997.
20. Rochat Y. and Taillard E.D. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics, 1:147–167, 1995.

Robust Engineering Design with Genetic Algorithms

Babak Forouraghi

Computer Science Department, Saint Joseph's University,
Philadelphia, PA, 19131, U.S.A.
bforoura@sju.edu

Abstract. This paper introduces a new robust optimization technique which performs tolerance and parameter design using a genetic algorithm. It is demonstrated how tolerances for control parameters can be specified while reducing the product's sensitivity to noise factors. As generations of solutions undergo standard genetic operations, new designs evolve, which exhibit several important characteristics. First, all control parameters in an evolved design are within a set of allowed tolerances; second, the resulting product response meets the target performance; and finally, the product response variance is minimal.

1 Introduction

Robust design of a product or a manufacturing process generally involves parameter design or design optimization and tolerance design, which is also known as sensitivity analysis [16]. The role of parameter design is to specify the levels of design variables (control factors) that minimize sensitivity of a product or process to uncontrollable noise factors such as manufacturing imperfections, component deterioration, and external operational conditions. During this step, tolerances are assumed to be wide enough as to keep the manufacturing cost low. If parameter design fails to produce adequately low functional variation of the product, then during tolerance design tolerances are selectively reduced based upon their cost effectiveness [17]. Generally, most design efforts are concentrated on system design, which involves innovation and the use of engineering sciences, and on tolerance design while ignoring parameter design [1, 2, 11, 18].

Deterministic approaches to quality improvement, such as Taguchi's method, allow designers to use a small number of experiments to determine effects of main control factors [16]. Incorporation of noise factors (tolerances) in design, on the other hand, often results in conducting a large number of experiments, which is an infeasible option in many design environments. An improvement to Taguchi's approach has been proposed, which is based on an interval-based, min-max optimization method [8]. This method has the advantage that it requires fewer experiments to perform sensitivity analysis, but it does rely on computationally intensive model-building methods, and it is not scalable to larger, more realistic design problems.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 562-572, 2004.

Other approaches to tolerance design include the heuristic techniques of search space smoothing and centers-of-gravity (CoG) [15]. These are iterative design centering techniques that use Monte Carlo simulation to draw large numbers of samples from the tolerance design region and then focus on the detailed features of the target area. The nominal design centers are then typically moved towards target areas until the desired yield is achieved. Although search space smoothing is computationally less intensive than CoG, both techniques require much auxiliary information in order to work properly. Furthermore, Monte Carlo simulations require large numbers of sampling points to assure high degrees of correlation between estimated and observed product yields [14].

Genetic algorithms (GAs) are another powerful search technique, which are modeled after natural genetics and have been applied to plethora of engineering design and optimization tasks [3, 7, 13]. The main strength of GAs is that, unlike many traditional optimization methods, they require minimal auxiliary information in order to exploit promising solution regions. For instance, gradient searches need derivatives in order to perform hill-climbing, but they fail when attempting to explore noisy, non-smooth, poorly-behaved search spaces. Dynamic programming methods need to maintain a global perspective function in order not to converge to local sub-optimum solutions, but they most often perform poorly when dealing with larger, real-world problems [12]. And finally, greedy combinatorial optimization methods require access to most if not all tabular parameters and are susceptible to converging to local sub-optimal solutions [13].

In the area of system reliability design GAs have been used in conjunction with interval programming [6, 19]. These approaches transform the interval programming aspect of the task at hand into an equivalent multiobjective optimization problem and then attempt to locate Pareto optimal solutions. For tolerance analysis, special types of GAs have been developed that use truncated Monte Carlo simulation in order to identify optimal allotment of tolerances [14]. The simultaneous use of GAs in both design optimization and tolerance design, however, is an issue that has not been investigated before.

In this paper we introduce a new approach to tolerance and parameter design using a genetic algorithm, whose main strength is twofold. First, our GA's chromosomal representation of the design space, unlike other proposed GA-based methods for design optimization, is interval-based. This feature allows designers to easily allot tolerances for design variables and simultaneously study effects of various tolerances on product performance indices. And second, the GA's method of evaluating candidate designs ensures convergence toward design regions where product responses are on target while showing minimal fluctuations.

The remainder of the paper is organized as follows. Section 2 provides a brief introduction to genetic algorithm and then discusses our approach to design optimization in more detail. Section 3 deals with optimum design of a temperature controller unit and reports on the simulation results. And finally, section 4 is the concluding remarks.

2 Chromosome Representation and Evaluation

Genetic algorithms (GAs) are powerful random-walk search techniques that have been successfully applied to a variety of engineering design and optimization tasks and are well explained in literature [3, 4, 7, 12, 13]. Our specific techniques for chromosomal representation and evaluation, however, are explained in this section.

In contrast to classical optimization methods, which support only ‘single’ point designs, the GA presented in this work builds tolerances for design parameters by successively evolving populations of hyper-rectangular design regions. For instance, traditionally a design vector X with n independent design parameters is represented as $(x_1,...,x_n)$. Here, however, the same design vector is represented as $(x_1^{center},..., x_2^{center})$ where the nominal values for each design variable are initially selected at random. Since the design tolerances are indicated a priori by designers, each input vector is internally represented by $([x_1^{low}, x_1^{high}], ..., [x_n^{low}, x_n^{high}])$, where x_i^{low} and x_i^{high} are the low-end and high-end of an interval for each design variable $x_i = (i=1,...,n)$, respectively. Consequently, solutions obtained in this manner no longer correspond to single points within the design space, but rather to hyper-rectangular regions. From the perspective of off-line quality control, this approach has the benefit of allowing designers to systematically incorporate into design the effects of uncontrollable variations that occur during the manufacture or normal operation of a product.

One final comment relates to the size of hyper-rectangular design regions and its effect on the produced product response. As discussed in Section 1, the primary concern in a robust design environment is with design of products that exhibit minimal scatter for a given range of input parameters. In an environment where design tolerances are selected by designers the genetic algorithm will attempt to find minimum scatter response regions which satisfy user-defined geometric constraints. The fitness assignment scheme developed in this work explores regions where a product's range of performance measure is deemed optimal. This point is further explained in the next section.

In terms of evaluating the fitness of a chromosome, it must be stated that as opposed to single point designs where every solution vector results in a particular product response, each candidate design in this formulation produces a range of responses. The implemented method for evaluating a chromosome, which essentially represents a hyper-rectangular region within the design space, utilizes fractional factorial experiments [16, 17]. Fig. 1 depicts an L_{18} experiment that can easily accommodate a design involving one 2-level and seven 3-level design variables.

Experiment	**Factors**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

Fig. 1. L_{18} ($2^1 \times 3^7$) experiment

A full factorial experiment for using six 3-level design parameters would typically require a total of 3^6 experiments. The L_{18} orthogonal array, on the other hand, would require only 18 experiments. Note that since the array can accommodate a total of 8 factors, the first 2 columns can easily be ignored. This type of design of experiments also involves application of statistical methods such as analysis of means and analysis of variance (ANOVA) in order to identify optimum design factor levels [17]. In the case of the GA, it was decided to use an L_{18} array to rank each design vector as follows. Each candidate design vector in a given population contains upper and lower bounds for each design parameter based on assigned tolerances. The levels 1, 2 and 3 associated with a particular design variable, therefore, correspond to the lower, middle and upper values of the interval associated with that particular factor. Each hyper-rectangular design region obtained in this manner produces a set of 18 responses.

The next step in evaluating designs is to assign absolute fitness values to each design candidate. Taguchi and other types of design of experiment methods routinely employ smaller-the-better, nominal-the-best or larger-the-better characteristic functions [16]. The best characteristic function for our case was nominal-the-best as it was desired to keep the design's performance level as close as possible to some predefined target value. Some modifications, however, were made in order to incorporate designers' requirements.

In particular, it was decided to use a variation of the mean-square error (MSE) function to assess the behavior of each design. Various forms of the MSE measure are widely used in many mechanical tolerancing design environments such as six sigma [10]. The appealing property of MSE is that not only it evaluates the central tendencies of some response region, but it also measures the amount of fluctuation that is present in that region. The modified MSE characteristic function used in our work was formulated as:

$$MSE = \frac{1}{n}\sum_{1}^{n} (y_i - y_T)^2 . \tag{1}$$

Here, n is the number design points associated with a particular hyper-rectangular design region (i.e., 18 for L_{18}), y_i is the produced response, and y_T is the desired target value. The GA's evaluation method can then use the inverse of MSE to assign higher fitness values to minimal-scatter designs that are centered at or near the desired target value.

After candidate designs in the current pool are assigned absolute fitness values in the manner described above, the genetic operations of reproduction, crossover and mutation [7, 13] follow. In each iteration of the algorithm the search for optimal solutions continues concurrently within many regions in the design space. This entire process of evaluation and selective breeding continues until some stopping criterion is met. In our case, the GA was allowed to evolve 30 generations of solutions, each containing 100 candidate designs. The behavior of the top ranking design along with the average behavior of the entire population can be monitored throughout the entire course of optimization. This will be discussed next where an actual design case is presented.

3 Design of a Temperature Controller

This section presents the optimum design of a temperature controller unit that has been previously used as a test case [8, 18].

The temperature controller circuit (see Fig. 2) involves 6 design factors: voltages E_1 and E_2; and resistances R_1, R_2, R_3, and R_4. The performance characteristic is the values of the resistance $R_\text{T-on}$ at which a relay activates the heater. The target value y^* for this characteristic is 3kΩ. The dependency relation for the performance characteristic is:

$$\frac{R_1 R_2 (E_2 R_4 + E_1 R_3)}{R_3 (E_2 R_4 + E_2 R_2 - E_1 R_2)} . \tag{2}$$

The design factors E_1, $E_{2,}$ R_1, R_2, R_3 and R_4 must vary within the ranges [7.5V, 9.5V], [4.5V, 7.5V], [0.5kΩ, 1.5kΩ], [6kΩ, 10kΩ], [2kΩ, 6kΩ], and [16kΩ, 48kΩ], respectively.

The main goal is to find the optimal design vector $X^* = (E_1, R_1, R_2, R_3, R_4, E_2)$ for the temperature controller such that the resulting performance is as close as possible to the target value of 3kΩ while minimizing fluctuations around that target value.

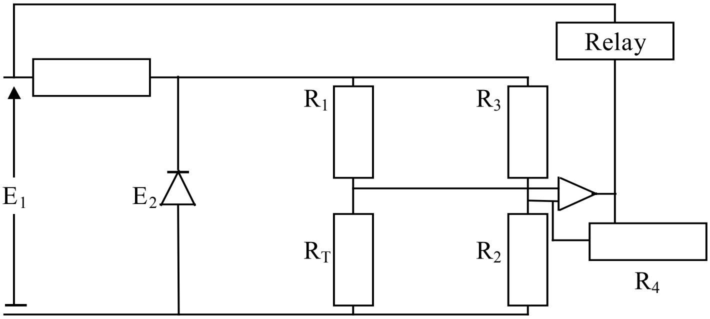


Fig. 2. Temperature controller circuit

The genetic algorithm produced a random pool of 100 candidate designs and individual designs were evaluated and ranked relative to one another based on how well they satisfied the particular performance index that was selected for this problem. Design chromosomes then underwent the genetic operation of selection whereby candidates to breed in the next generation were chosen. The standard genetic operations of crossover and mutation were applied to the entire pool of the selected individual solutions with probabilities of 1.0 and 0.001, respectively. The algorithm was allowed to run for 30 generations.

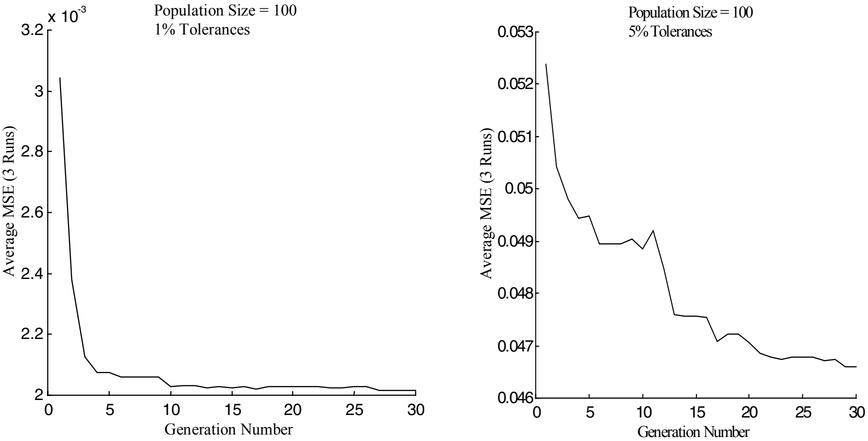


Fig. 3. Average MSEs (1% and 5% tolerances)

A close examination of fitnesses of individual members of a population during the course of the search reveals the GA's direction of informed walk within the solution space. To this end, the performance index of each generation's best design was recorded. To account for statistical fluctuations present in selection of the initial pool and also in applying the genetic operators of crossover and mutation, each experiment was repeated three times for a given user-defined tolerance level. The performance profile of 30 generations of designs using 1% and 5% design tolerances is depicted in Fig. 3.

As seen in Fig. 3, the initial members of the pool perform rather poorly in satisfying the design constraints. This implies that the generated designs do not achieve the target performance, or they do that while exhibiting large degrees of variation. After each application of the genetic algorithm, the general fitness of the entire pool tends to increase toward more acceptable levels.

Although 5% tolerances are acceptable in many typical applications, for further testing it was decided to test the true mettle of the GA by using wider tolerances of 10% and 20%. Results of these experiments, which depicted similar patterns, are shown in Fig. 4.

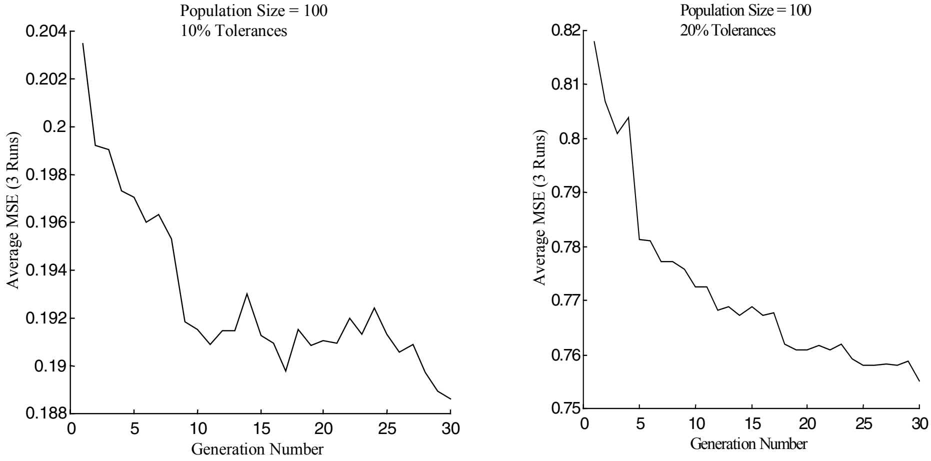


Fig. 4. Average MSEs (10% and 20% tolerances)

Initially, the product response for 10% and 20% tolerances is poor but with each application of the genetic algorithm the quality of newly evolved designs improves. Clearly, the amount of fluctuation around the controller's response target value has to increase in order to accommodate higher design tolerances. This observation directly explains as to why the average MSE of an entire population grows larger as the user requested tolerances become wider.

Another useful experiment for assessing the behavior of a genetic algorithm is to examine not only the highest-ranking design in the population, but also the general tendencies of an entire population. This type of observation helps to understand whether the entire pool of designs is performing hill-climbing or just a few randomly selected elite individuals are exploring optimal response regions. Fig. 5 shows the results of 3 statistically independent runs of the algorithm.

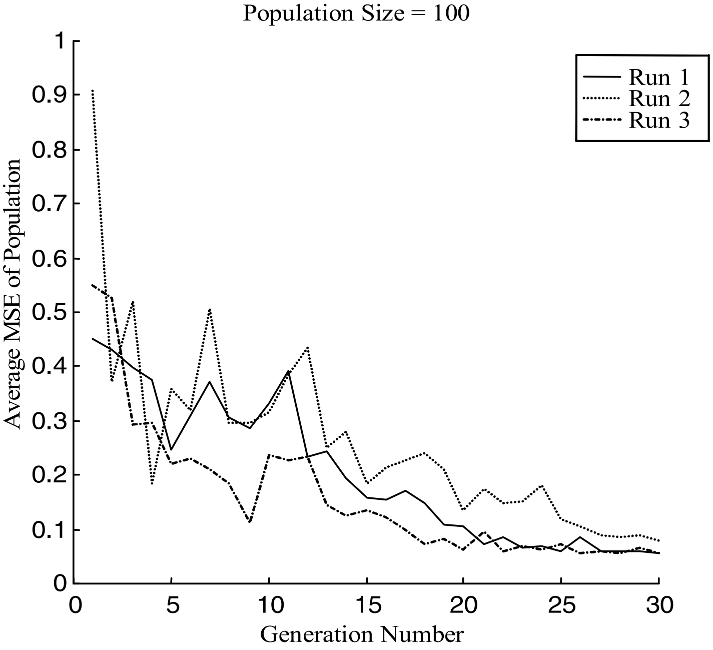


Fig. 5. Average MSE of an entire population over 3 independent iterations of the algorithm (5% tolerances)

The average behavior of an entire population in each run of the algorithm follows a similar pattern. Initial members of the pool perform poorly while future applications of genetic operators redirect the search toward more promising (fitter) regions of the response surface. The fluctuations that appear in the 3 fitness plateaus are directly attributed to the non-deterministic nature of the algorithm in which the crossover and mutation operators can introduce poorer traits into a generation. This genetic drifting, however, typically disappears in subsequent generations as fitter individuals dominate the population [7].

Having considered individual as well as collective behavior of the 100 candidate designs, Fig. 6 demonstrates the quality of the achieved designs after 30 evolutions of an original random pool. The first row in Fig. 6 shows the solution obtained by using a traditional min-max optimization method [8]. The optimum design vector x^* = (8.31, 0.64, 9.27, 2.06, 34.55, 7.18) using a 5% tolerance level results in the value of 3.23±0.36kΩ for the performance characteristic $R_{\text{T-on}}$. In order to better describe the quality of obtained designs, it was decided to measure the coefficient of variation (*CV*) for each design, which is the ratio of the standard deviation to the mean expressed in percentage terms:

$$CV = \frac{s}{\overline{y}} \cdot 100 \ . \tag{3}$$

Design Method	Design Tolerances	Design Parameters						Response $\mathbf{R}_{\text{T-on}}$	Variation Coefficient
		E_1	R_1	R_2	R_3	R_4	E_2		
Min-Max	5%	8.31	0.64	9.27	2.06	34.55	7.18	3.23±0.36	11.2%
GA	1%	8.44	1.41	6.50	3.70	32.29	6.62	2.99±0.03	1.2%
	1%	8.53	1.09	8.82	4.01	38.23	6.35	2.98±0.03	1.1%
	1%	7.56	1.31	9.73	5.04	44.24	6.35	2.99±0.04	1.3%
	5%	7.80	1.28	7.70	3.68	43.07	7.06	3.04±0.18	6.0%
	5%	8.50	0.86	7.88	2.68	41.51	6.19	2.96±0.19	6.7%
	5%	8.31	1.36	8.13	4.27	41.57	7.38	2.97±0.17	5.6%
	10%	7.97	1.22	6.73	3.11	40.73	7.32	2.91±0.38	13.0%
	10%	7.61	1.32	7.61	3.83	41.68	7.13	2.93±0.38	13.0%
	10%	7.80	1.17	7.79	3.49	43.38	7.17	2.98±0.36	12.3%
	20%	7.87	1.22	6.25	3.18	43.94	7.29	2.70±0.72	26.4%
	20%	7.67	1.42	6.90	4.21	45.57	7.26	2.50±0.61	24.6%
	20%	7.68	1.32	6.85	3.68	46.59	7.23	2.60±0.64	24.6%

Fig. 6. Optimum designs for the temperature controller

The rationale for using this indicator is that *CV* is a more descriptive tool than the standard deviation *s* alone as it describes *s* as a proportion of the mean [5, 9]. Consequently, the *CV* measure of 11.1% for the min-max solution (row 1 in Fig. 6) is rather high considering the fact that the response value of 3.23 is not as close to the target value of 3kΩ as it is required by design constraints.

The second row in Fig. 6 depicts a number of solutions obtained by our genetic algorithm. For each requested tolerance level, there are included 3 designs obtained from 3 statistically independent experiments. For instance, the first design with 5% tolerance levels identifies the optimum design vector (7.80, 1.28, 7.70, 3.68, 43.07, 7.06) which results in $R_{\text{T-on}}$ = 3.04±0.18 with a CV measure of 6.0%. Comparison of our result to that obtained via the min-max method implies the following observations. First, the GA solution is far superior in satisfying the design requirements by producing a design whose response is extremely close to designers' requested target value, and also, the resulting fluctuations around the response are quite acceptable considering the wide design margin of 5%. And second, the non-deterministic nature of the GA allows designers to inspect various promising regions in the solution space with minimal effort. This point is best demonstrated by examining the remaining 2 solutions that were obtained for 5% tolerance designs.

Furthermore, other reported designs in Fig. 6 that pertain to higher tolerance designs of 10% and 20% consistently provide designers with robust designs where the responses are either on target or very close to it. Although these designs, which do allow higher tolerances, exhibit larger *CV* values, this is to be expected as wider margins in design variables naturally entail higher response fluctuations.

Another advantage of GA's approach to mechanical design tolerancing is the non-deterministic nature of the algorithm. In many instances designers face situations where either due to manufacturing and operational constraints or economic factors a given design scenario can no longer remain optimal (e.g. a new motor may not provide the same range of cutting speeds). In these cases, a GA will be able to provide designers with myriad of optimal, tradeoff solutions. The deterministic approaches of traditional min-max optimization, response surface methodology or Taguchi's experimental design, however, are rigid in that they do not easily allow incorporation of model uncertainty parameters into design.

4 Conclusions

Improving production quality and reducing cost while performing tolerance design is an inevitable challenge in today's competitive global market. Non-deterministic optimization methods such as genetic algorithms are powerful random-walk search techniques that allow designers efficiently explore many optimal regions of a design space in parallel.

In this paper we discussed a new approach to quality control and tolerance design using genetic algorithms. The main strength of the genetic algorithm presented here is in its use of biologically motivated concepts of crossover and mutation as to evolve populations of optimal designs. Specifically, it was demonstrated how the GA could be used to identify optimal hyper-rectangular regions within the design space where the resulting response of a temperature controller unit was fixed on a target value while exhibiting minimal variations. It was demonstrated that the non-deterministic nature of GAs allows designers to inspect many promising design regions within which a product can operate optimally and within anticipated margins.

References

1. Craig, M.: Limits of Tolerance. Manufacturing Engineer, 6 (1996) 139-143
2. Creveling, C.M.: Tolerance Design: A Handbook for Developing Optimal Specifications. Addison Wesley, New York (1997)
3. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, New Jersey (1995)
4. Forrest, S.: Genetic Algorithms. ACM Computing Surveys. 28, 1 (1996) 77-80
5. Freund, R.J., Wilson, W.J.: Statistical Methods. Academic Press (1997)
6. Gen, M., Cheng, R.: Optimal Design of System Reliability Using Interval Programming and Genetic Algorithms. Computers and Industrial Engineering, 31, 1 (1996) 237-240

7. Goldberg, D.: Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications. Wiley & Sons, New York (1998)
8. Hadjihassan, S., Walter, E., Pronzato, E.: Quality Improvement via Optimization of Tolerance Intervals during the Design Stage. In Kearfott and Kreinovich (eds.): Applications of Interval Computations. Kluwer Academic Publishers, California (1996) 91-131
9. Harris, R.J.: A Primer of Multivariate Statistics. Academic Press, Inc. (1985)
10. Harry, M.J., Stewart, R.: Six Sigma Mechanical Design Tolerancing. Motorola University Press, (1988)
11. Ishibuchi, H., Tanaka, H.: Multiobjective Programming in Optimization of the Interval Objective Function. European Journal of Operations Research, 48 (1990) 219-225
12. Karr, C.L.: Genetic Algorithms for Modeling, Design and Process Control. Communications of ACM, 3 (1993) 233-238
13. Koza, J., Bennett, F.H., Andre, D., Keane, M.A.: Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann, California (1999)
14. Lee, J., Johnson, J.E.: Optimal Tolerance Allotment Using a Genetic Algorithms and Truncated Monte Carlo Simulation. Computer-Aided Design, 25, 9 (1993) 601-611
15. Liu, Y., Foo, S.W.: Fast Search Algorithm for Tolerance Design. IEE Proceedings of Circuits Devices and Systems, 45, 1 (1998) pp. 19-23
16. Peace, G.S.: Taguchi Methods: A Hands-On Approach. Addison Wesley (1993)
17. Phadke, M.: Quality Engineering Using Robust Design. Prentice Hall, New York (1989)
18. Taguchi, M., Phadke, M.S.: Quality Engineering Through Design Optimization. IEEE Global Telecommunications Conference, Atlanta, GA (1984) 1106-1113
19. Yokota, T., Gen, M.: Optimal Interval Design for System Reliability with Incomplete FDS by Means of Improved Genetic Algorithms. Electronics and Communications in Japan, 81, 1 (1998) 84-94

Evolutionary Computation Using Island Populations in Time

Ben Prime and Tim Hendtlass

Centre for Intelligent Systems and Complex Processes
School of Information Technology
Swinburne University of Technology
{bprime, thendtlass}@swin.edu.au

Abstract. This paper demonstrates that the reintroduction of genetic information from past generations can be used to alter the genetic diversity and convergence characteristics of an evolutionary algorithm (EA). Traditional EAs breed a new generation from the preceding *generation*: in this work a new generation is bred from the preceding *generations*. The gene reuse scheme described in this paper allows control of the genetic diversity of the population with a resulting beneficial effect on the convergence performance.

Keywords: Genetic algorithm, evolutionary algorithm.

1 Introduction

The Evolutionary Algorithm (EA) is a robust optimisation technique that is based on Darwinian evolution [1,2]. Although the artificial representation of genetic structure is quite rudimentary, this method can be used to find solutions to a plethora of problems. The procedure uses Darwinian selection to cause a 'population' of possible answers to progress (or *evolve*) towards better optimal points. This is done through successive generations of evolution where new potential solutions are 'bred' from the previous generation, with better performing solutions being rewarded with a higher probability of passing on their own genetic material to future generations. Individual solutions consist of a chromosome of genes, and it is the values of these that specify the particular solution represented by this chromosome. Typical EA implementations tend to winnow this genetic information as they progress from one generation to the next.

As pieces of solutions are recombined with each other (or copied) and mutated, the EA causes the population as a whole to 'explore' the problem, particularly concentrating on regions with promisingly high fitness. This concentration of effort is a result of the predominance in the population of particular gene values associated with higher fitness regions. One of the unfortunate drawbacks of the EA is that it is not guaranteed to find the global optimum for every problem. A particular set of gene values may so dominate a population that every individual chromosome describes a very similar solution within a small region of problem space. When this happens the EA is said to have converged. Premature convergence is said to have occurred if the solution converged on is sub-optimal [3,4]. With no diversity in gene values explora-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 573-582, 2004.

tion outside the current region is highly improbable (unless extreme mutation is allowed).

Since escape from premature convergence is extremely hard, various strategies have been undertaken in order to avoid premature convergence. These include the use of island population schemes, cyclic mutation, tabu mechanisms and multiploid structures. Each of these approaches attempts to preserve diversity in gene values without inhibiting the general focus of the search toward regions of known merit.

It has been shown that the performance of the EA can be improved by means of the concept of using "islands" [5,6]. Island populations can be thought of as dividing an entire population of an EA into physically separate (or caste) parts. It has been noted that under such circumstances each group within the total population will evolve separately. Each population is loosely linked with the others, by the occasional transfer of entire individual solutions between groups. The advantage of this method does not become apparent until differing conditions, such as the imposition of varying probabilities of mutation are applied to each population. For further discussion of this see, for example, [7].

High mutation has the effect of significantly increasing the diversity of gene values but, being a stochastic process, may interfere with the guided exploration of problem space. It may produce movements that are inconsistent with the (unknown) scale of the significant features in problem space. Cyclic mutation addresses these concerns by the periodic variation of mutation parameters (probability and/or maximum magnitude) throughout the evolution process [8]. High mutation periods force an increase in diversity and are followed by periods of low mutation that allow the exploration of the problem space now occupied.

Tabu search maintains a list of areas of the solution space that have already been searched [9]. When used in collaboration with an evolutionary algorithm the breeding process is modified so that only solutions not already on this tabu list are allowed to be bred. This pushes future generations into areas that have been less thoroughly investigated. For a non-quantized problem space, the tabu list can be replaced by a special neural network that stores an ever-evolving series of pointers to regions of known good and bad points. Then the probability of a new solution being accepted into the population is inversely proportional to the distance from the closest of these points. It is important that the probability coefficients differ between good and bad points [10] as both of these directly prevent convergence (premature or otherwise).

The work reported in this paper uses two island populations, one of which is conventional and contains the current population. The other island contains gene values taken from earlier populations, essentially an island in time. This will be referred to as storage. The gene values (not whole chromosomes) stored cannot interbreed or mutate. They are just held for possible reintroduction into a future generation: that is, the "reincarnation" of old genetic material into a later generation. The aim is to reintroduce information in order to preserve useful diversity, where useful diversity is that which has a beneficial effect on convergence by allowing backtracking in the path to the solution. Backtracking when premature convergence is detected should improve the performance of the search.

Acan and Tekol have reported on chromosome reuse [11]. In their work, after the breeding process, individuals from some fitter part of the population that have not been selected as a parent are copied into extra storage. At the next breeding process these individuals compete equally with the members of the current conventional population for parenting opportunity. An upper limit is placed on the size of this auxiliary storage, once this is filled only individuals with a fitness higher than the least fit individual currently stored there can be added and replace the least fit individual.

Gene, rather than whole chromosome, reuse can be accomplished by the use of multiploid structures in which each gene has its own restricted array of values, only one of which can be expressed at any particular time. For further information see, for example, [12].

This work involves a more extended process than either of the above in which a targeted selection is made from all extinct gene values for deliberative rather than stochastic reintroduction into the current population. It differs from other work in the use of replacement at the gene level, in the range of values available to be chosen and the deliberative nature of the reinsertion mechanism.

2 Diversity

A numeric measure of diversity is required in order to discuss changes in diversity. Integer gene values were used in this work and a suitable definition for the diversity D_G of gene G is:

$$D_G=(V_G-1)/(P-1) \tag{1}$$

where V_G is the number of different values occurring in gene G in the P individuals in the population (it is independent of the distribution of these values). Note that D_G is a simple measure that only reflects the number of values and does not in any way reflect how useful these values might be for any particular problem.

At the time of initialisation an EA starts with random gene values (assuming no seeding arising from *a priori* knowledge of the problem). The diversity of the population at this point in time is quite large, and results in a sparse but general sampling of the solution space. As the EA progresses the function of the reproduction operators are inclined to cause each of the (formerly random) potential solutions to become more similar as they move towards optima within the solution space. The sampling of optimal regions increases as individuals in the population become more similar, however the scope of exploration decreases. In the case of premature convergence the population loses diversity to the point where further search is highly improbable. Once diversity has reached its low point the search tends to become a “creep mutation” search in the sole remaining area of interest. This is characterized by a thorough search of a small area with little regard for the overall problem.

Figure 1 shows the variation in diversity with time within a population without mutation and in the absence of a fitness gradient. It also shows the result of a very simple scheme for reintroducing extinct gene values. After each generation had been bred, the gene values that had become extinct (found by comparing the old and new populations) were placed into storage. The average occurrence rate of the gene values in the new population is calculated and those gene values whose number of occurrences exceeds this are noted. For each of these over-represented gene values on average one in four instances are replaced by values drawn randomly from storage to reduce the occurrence of the over-represented gene value towards the average. No further material was added to storage after generation 50, instead one item was permanently removed from storage each generation after that.

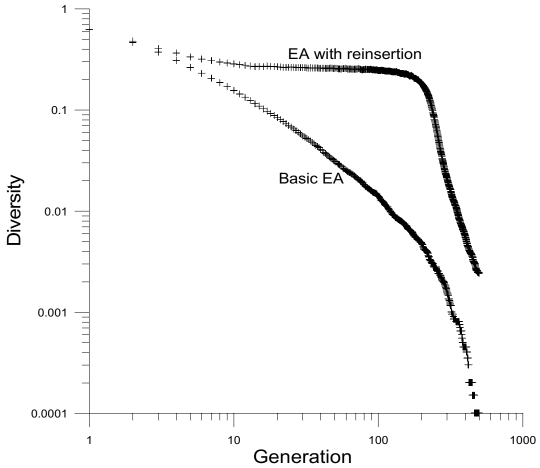


Fig. 1. The change in diversity in a region of uniform fitness with and without the reintroduction of previously lost genetic material. After 50 generations no further material was added to storage. The sharp fall in diversity at 200 generations for the EA with reinsertion arises from the emptying of the storage mechanism.

Note the rapid decrease in diversity without a mechanism for reintroducing extinct values. Using the reintroduction mechanism described above, initially the diversity drops more slowly. However after 200 generations, as the storage was deliberately emptied, the rate of diversity loss approaches that experienced by the basic evolutionary algorithm without reinsertion of genetic material. In this case, without a fitness gradient, one reintroduced value is as good (or bad) as any other. When a fitness gradient exists, to be useful naturally the values returned must show a bias towards values that assist solution to the problem.

3 The Basic Evolutionary Algorithm

The basic evolutionary algorithm used is conventional as follows.

1. An initial generation of solutions is created with randomly chosen gene values. The fitness of each individual in this population is calculated.
2. Create a new generation by crossing over two (or more) parents to form each new individual. The probability of selecting a particular individual to be a parent is proportional to its relative fitness compared to the rest of this generation. After the new individual has been produced, it may be mutated by small random adjustments to its gene values.
3. Calculate the fitness of each member of the new generation. Unless some stop criterion has been reached (i.e. the fitness of the best performing individual is better than some threshold or a predetermined number of generations has been bred) loop back to step 2 and continue.

While there are many variations in the exact way that parents are selected and crossover preformed, these are basically ways of fine-tuning the algorithm to improve its performance on a particular problem. This paper is concerned with the effect of reintroducing genetic material on the basic EA, there is no reason to suppose that the use of any of the many variations known to the authors would significantly affect the results obtained.

4 The Genetic Material Reinsertion Algorithm

The reinsertion of genetic material algorithm used interposes an extra step between steps 2 and 3 above. This step can be expressed as two sub steps as follows.

i. After a new generation of individuals has been conventionally bred it is compared with the old generation. For each gene, a list is made containing all the values of this gene that have become extinct as a result of this breeding process. These values are added to storage, if not already there.

ii. For each gene in the chromosome, a list is made of the gene value distribution in the new population. Gene values that occur more frequently than average are identified and a randomly chosen proportion of these are replaced by values retrieved from storage. In the work reported here the number of a particular gene value replaced is one quarter of the difference between its number of occurrences and the mean number of occurrences. The value used for a particular replacement operation was chosen from storage by one of two methods:

```
a) Random insertion (RI):
    Replace all instances of a gene value identified for replacement with
    one randomly select from storage.
or
b) Tested insertion (TI):
    Repeat
        Randomly chose a value from the freezer.
        Perform a trial insertion and measure the change in fitness of the
        recipient, $\Delta f$.
        Accept this insertion with a probability P =sigmoid($\Delta f$).
    Until an insertion is accepted or N randomly chosen values have been
    tried.
```

5 Methodology

The aim of this work is to investigate the effect of reinsertion of genetic material on the incidence of premature convergence. Hence experiments were performed on two functions that have a high probability of exhibiting premature convergence. Additionally, the normal mutation parameters were chosen to increase the probability of premature convergence, although not to the extent that successful completion failed to occur a significant proportion of the time.

The first fitness function used is given in two-dimensional form in equations 2 and 3, in which G_1 and G_2 are the two genes values.

$$1+(T)^{0.25}-\cos(5\pi(\sqrt{T})) \tag{2}$$

$$\text{where for two dimensions } T=(G_1-0.1)^2+(G_2-0.2)^2 \tag{3}$$

$$\text{and for three dimensions } T=(G_1-0.1)^2+(G_2-0.2)^2+(G_3-0.3)^2 \tag{4}$$

Figure 2 shows the two-dimensional fitness surface produced consisting of a series of deep concentric rings (with a separation of 0.4) superimposed on a shallow dish. The global minimum at (0.1,0.2) has a value of zero, the next ring has a minimum value of 0.632 and the ring after that 0.894.

Similar functions can be readily constructed with any required number of dimensions, the difficulty of solving it increasing rapidly as the number of dimensions increases. In this work a 3D version (with a global minimum at (0.1,0.2,0.3)) produced by equations 2 and 4 was also used.

The second fitness function, given in n dimensional form by equation (5)

$$\sqrt{\sqrt{\sum_{i=1}^{n}(X_i+8)^2}+0.1*\sqrt{\sum_{i=1}^{n}(X_i+2)^2}+0.2*\sqrt{\sum_{i=1}^{n}(X_1-3)^2}} \tag{5}$$

produced the two-dimensional fitness surface shown in figure 3. Irrespective of the value of *n* this function has a single minimum with a value of zero when each gene value is 3. This is the leftmost of the three minima in figure 3. This function was chosen as it produces a fitness surface that is quite different to the radial surface produced by equation 2.

The evolutionary algorithm used was deliberately simple without a local heuristic to better reveal the effect of the reintroduction of genetic material. Crossover was simple one point cross over and the mutation probability was 0.2. The mutation magnitudes used came from a Gaussian distribution around zero with a standard deviation of 0.08, a value chosen to make finding the global minimum difficult but not impossible.

All results quoted are the averages from 1000 independent repeats with different randomly chosen starting values. Each run was continued for a fixed number of generations after which preliminary work had shown that no further significant change resulted. For the first function this was 500 generations (2D) or 2000 generations (3D), for the second this was 10,000 generations. The maximum number of repeat trial insertions allowed when using tested reinsertion was ten.

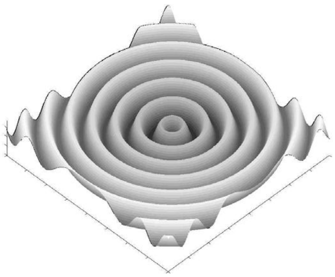


Fig. 2. The fitness surface produced by equations 2 and 3 over the range ±2 on each axis

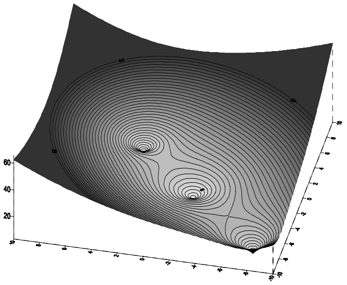


Fig. 3. The fitness surface produced by equation 5 when *n*=2 over the range ±10 on each axis.

6 Results

For the first function the results are shown in figures 4 to 11.

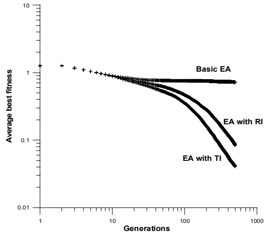


Fig. 4. First function in 2D. Best fitness per generation. Average over 1000 repeats. Global optimum fitness zero.

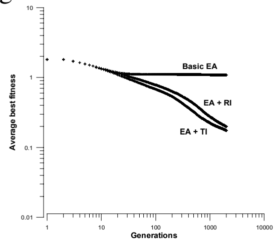


Fig. 5. First function in 3D. Best fitness per generation. Average over 1000 repeats. Global optimum fitness zero.

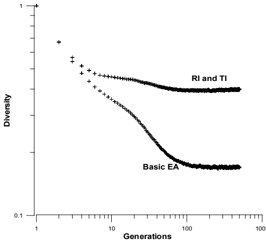


Fig. 6. First function in 2D. Diversity per generation. Average over 1000 repeats.

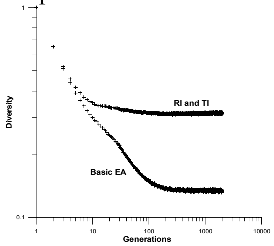


Fig. 7. First function in 3D. Diversity per generation. Average over 1000 repeats.

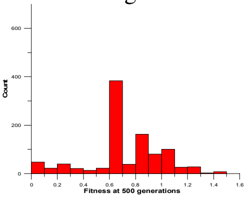


Fig. 8. First function in 2D, basic EA. Distribution of fitness values at 500 generations over the 1000 repeats.

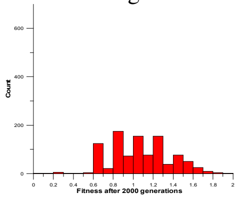


Fig. 9. First function in 3D, basic EA. Distribution of fitness values at 2000 generations over the 1000 repeats.

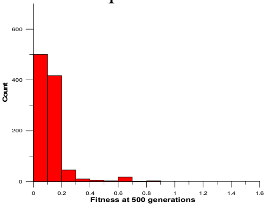


Fig. 10. First function in 2D, basic EA with random reinsertion. Distribution of fitness values at 500 generations over the 1000 repeats.

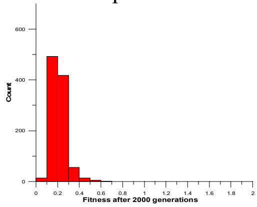


Fig. 11. First function in 3D, basic EA with random reinsertion. Distribution of fitness values at 2000 generations over the 1000 repeats.

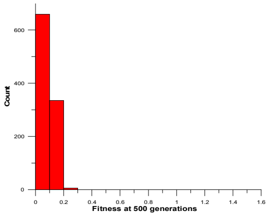


Fig. 12. First function in 2D, basic EA with tested reinsertion. Distribution of fitness values at 500 generations over the 1000 repeats.

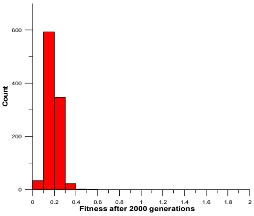


Fig. 13. First function in 3D, basic EA with tested reinsertion. Distribution of fitness values at 2000 generations over the 1000 repeats.

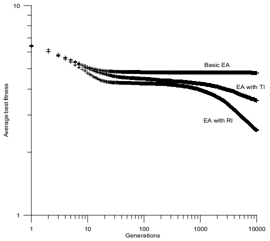


Fig. 14. Second function in 2D. Best average fitness per generation. Average over 1000 repeats. Global optimum fitness zero.

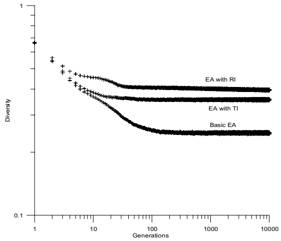


Fig. 15. Second function in 2D. Diversity per generation. Average over 1000 repeats

Figure 4 shows the average best fitness for each generation for the 2D version of the first function and figure 5 the average best fitness for each generation for the 3D version of this function. Figures 6 and 7 show the average diversity for each generation for the 2D and 3D versions respectively. The next two figures show histograms of the fitness found from the 1000 runs for the basic EA, figure 8 is for the 2D version (at 500 generations) and figure 9 for the 3D version (at 2000 generations). Figures 10 and 11 show this information for the basic EA plus random reinsertion and figures 12 and 13 show this information for the basic EA plus tested reinsertion. To facilitate comparison the same vertical axes scale has been used for all the histograms.

Figures 14 and 15 show the average fitness and diversity for the second function (equation 5) in two dimensions. The same general performance advantage can be seen, indicating that the advantage does not come from some particular aspect of the first fitness function, although random insertion is this time better than tested insertion.

7 Discussion

The figures above show that reinsertion of genetic material does significantly improve the performance of an evolutionary algorithm on two different problems known to be prone to premature convergence. Both the random reinsertion and tested reinsertion algorithms show a significant improvement over the basic EA. At first sight this seems surprising, as there is no guarantee that the reinserted material is immediately beneficial. However, such a requirement is a desirable but not necessary condition, as the normal evolutionary pressure will in time remove the disadvantageous.

Both insertion algorithms first identified all over represented gene values in the population and then identified which of these instances should be replaced. For random reinsertion all the instances of each specific gene value so identified were then replaced by the same randomly chosen value. Other experiments proved this to be more advantageous than replacing each of the designated instances with an independently randomly chosen value from storage. This is again believed to reflect the difference between an immediate effect (which therefore reflects in the breeding probability of the individual that received the insertion) and the effect over a more extended number of generations. By inserting multiple instances of the randomly chosen value it has longer to prove its worth before facing extinction.

The tested reinsertion algorithm effectively applied a first filter to the reinserted material, thus leaving less ‘cleaning up’ for the evolution to do. Since immediate and longer-term benefits are two different (if overlapping) requirements, this first filter should not be too drastic. For this reason, the probabilistic filter used here has been designed so that, although biased towards accepting immediately beneficial insertions, it allows for the possibility of accepting an immediately detrimental replacement. As it was more probable that the inserted value was beneficial in the short term, the value was more likely to be passed onto future generations as a result of the breeding process even though immediately after insertion it only once occurred in the population. Interestingly there was no clear winner of the two reinsertion algorithms as a result of the work reported in this paper. For this simple fitness function, both the algorithms approximately tripled the computational load per generation, this being partly compensated for by the reduced number of generations required.

8 Conclusion

The results from this first investigation clearly show that reinserting genetic material can be advantageous, at least for the two problems described in this paper. The additional added computational expense is modest for the RI algorithm as this requires no more evaluations per generation than the basic EA. The increase in computation is somewhat more for the TI algorithm mainly due to the larger number of evaluations required and may or may not lead to a performance improvement.

Other work suggests that the results may well be applicable to other problems for which a conventional evolutionary algorithm exhibits a tendency to converge in suboptimal local minima. Further work will be necessary to uncover how critical (and possibly problem specific) the choice of methods for selecting values to be placed into and recalled from storage are.

Acknowledgment. The authors gratefully acknowledge contributions made by Clinton Woodward.

References

[1] J H Holland “Adaption in Natural and Artificial Systems” University of Michigan Press 1975.

[2] Davis, Lawrence, ed. "Handbook of Genetic Algorithms". first ed. New York: Von Nostrand Reinhold. 1991.

[3] Mauldin, M. L. "Maintaining diversity in genetic search." In Proceedings of the National Conference on Artificial Intelligence, 247-50. 1984.

[4] Goldberg, D E. 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley.

[5] R Tanese "Distributed Genetic Algorithms" Proceedings of the Third International Conference on Genetic Algorithms, George Mason University 1989.

[6] D Levine. "A parallel genetic algorithm for the set partitioning problem" Technical report ANL-94/23, Argonne National Library 1994.

[7] H Copland and T Hendtlass "Migration Through Mutation Space: A Means of Accelerating Convergence", Proceedings of ICANNGA97, England, 1997.

[8] T Hendtlass "On the use of variable mutation in an evolutionary algorithm," Proceedings of IEA/AIE-97, USA, Gordon and Breach. 1997.

[9] F Glover. "A Users Guide to Tabu Search" Annals of Operations Research Vol 41 J.C Baltzer AG, 1993.

[10] J Podlena and T Hendtlass "An Accelerated Genetic Algorithm", Applied Intelligence, Kluwer Academic Publishers. Volume 8 Number 2, 1998.

[11] A Acan and Y Tekol "Chromosome Reuse in Genetic Algorithms", Lecture Notes in Computer Science, Vol 2723, pps 695-705, Springer Berlin, 2003.

[12] Clinton Woodward and Tim Hendtlass**.** "Dynamic Trait Expression for Multiploid Individuals of Evolutionary Algorithms" Lecture Notes in Artificial Intelligence, Vol. 2070 pages 374-382 Springer, Berlin, 2001.

Genetic Algorithm Based Parameter Tuning of Adaptive LQR-Repetitive Controllers with Application to Uninterruptible Power Supply Systems

Mahdi Jalili-Kharaajoo[1], Behzad Moshiri[2], Karam Shabani[2] and Hassan Ebrahimirad[1]

[1] Young Researchers Club
Azad University, Tehran, Iran
{mahdijalili, h.ebrahiimirad}@ece.ut.ac.ir
[2] CIPCE, ECE Dep., University of Tehran, Iran
{moshiri, shabani}@ut.ac.ir

Abstract. In this paper, an adaptive Linear Quadratic Regulator (LQR) with Repetitive (RP) control is applied to Uninterruptible Power Supply (UPS) systems. The RP controller with forgetting parameters is used to attenuate the effects of periodic disturbances. In order to achieve an output signal with law Total Harmonic Distortion (TDH), the parameters of RP controller should be tuned excellent. In the paper, using genetic algorithms optimization, these parameters are optimal tuned. Simulation results of the closed-loop system with proposed controller reveal the effectiveness of the proposed approach.

1 Introduction

Uninterruptible Power Supplies (UPS) systems, which supply emergency power in case of utility power failures, are used to interface critical loads such as computers and communication equipment to a utility power grid. Many discrete time controllers have been designed to control a single-phase inverter for use in UPS [1-5]. In order to apply advanced control techniques in low coat micro controllers, a proper choice of the controller gains can be made off-line, which simplifies the control algorithm.

In this paper, an adaptive LQR with repetitive controller (LQR-RP) for single-phase UPS application is designed. In the proposed controller, a RLS estimator identifies the plant parameters, which are used to compute the LQR gains periodically [6]. The quadratic cost function parameters are chosen in order to reduce the energy of the control signal. The repetitive control action improves the response of the controller mainly in the presence of cyclic loads [7]. In [5] it is shown that using the repetitive controller with forgetting exponential coefficients, the control results can be improved significantly, which leads to an output voltage with low Total Hormonic Distortion (THD). There is not any efficient analytic mechanism to obtain the optimal values of the parameters of theses forgetting coefficients. So, we have to use intelligent optimization methods. Genetic Algorithms (GAs) [8] is one the most efficient methods for this application. In this paper, using GAs the optimal tuning of repetitive controller's parameters is performed. Simulation results on the closed-loop

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 583-593, 2004.

system with obtained parameters conform the ability of the proposed method to achieve an output signal with low and admissible THD.

2 Model of the Plant

The conventional single-phase PWM inverter is shown in Fig.1 [9] The *LC* filter and the resistive load *R* are considered to be the plant of the system. The power switches are tuned on and off during each interval T_s such that V_{in} is a voltage pulse of magnitude *E,0,-E* and a width ΔT . The plant can be modeled by the state space variables V_c and i_L

$$\begin{bmatrix} \dot{V}c \\ iL \end{bmatrix} = \begin{bmatrix} -1/RC & 1/C \\ -1/l & 0 \end{bmatrix} \begin{bmatrix} Vc \\ iL \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} u \ , \ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} Vc \\ iL \end{bmatrix} \tag{1}$$

Then, a discrete time model of the plant with a Zero Order Hold (ZOH) and a sample period T_s is given by

$$\begin{matrix} x(k+1) = A_d x(k) + B_d u(k) \\ y(k) = C_d x(k) \end{matrix} ; \begin{matrix} x(k) = [V_c(k) \quad i_L(k)]^T, A_d = e^{AT_s} \\ B_d = A^{-1}(e^{AT_s} - I)B, C_d = C \end{matrix} \tag{2}$$

To convert the control signal *u(k)* calculated by the digital processor into a pulse width signal, the following equation is used

$$\Delta T(k) = \frac{|u(k)|Ts}{E} \tag{3}$$

The pulse amplitude will have the same sign of the control signal *u(k)* and the maximum width theoretically available in a sample period is $\Delta T = T_s$ when $|u(k)| = E$. In practice, ΔT is lower than T_s, because there is a loss in duty cycle during the dead time of the inverter switching.

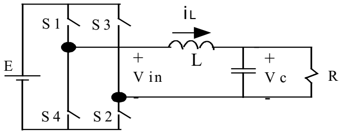


Fig.1. Inverter, filter and load.

3 Adaptive LQR-RP Controller Design

The servo-system with integrator shown in Fig. 2 is proposed [5]. It has the objective of tracking the discrete sinusoidal *r(k)* reference in each sample instant. The system output *y(k)* is the capacitor voltage in the discrete form $V_c(k)$. The total control signal *u(k)* is obtained adding the LQR action with the repetitive action. The LQR has the objective to minimize a cost function defined a priori. The RP has the objective to reduce the steady state error due to periodic disturbances.

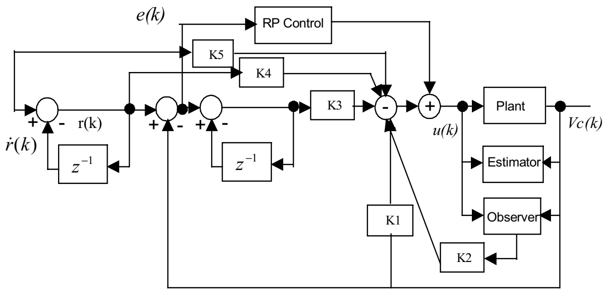


Fig.2. Block diagram of the closed loop system.

3.1 LQR Controller

The state variables used in the LQR are the measured output voltage *V$_c$(k),* the estimated inductor current $\hat{i}_L(k)$, the integrated tracking error e(*k*), all with a feedback action and the discrete reference *r(k)* and its derivative $\dot{r}(k)$ with a feed forward action. Each state variable has a weighting K_i tuned in function of $\theta(k)$. The state vector $\varpi(k)$ is defined as

$$\varpi(k) = [V_C(k)\ \ i_L(k)\ \ V(k)\ \ \ r(k)\ \ \ r_d(k)]^T \tag{4}$$

and the LQR control signal is given by

$$u_{LQR}(k) = -K\varpi(k) \tag{5}$$

To design the optimal gains $K_1, K_2, \ldots, K_5$, the system must be represented in the form

$$\varpi(k+1) = G\varpi(k) + Hu_{LQR}(k) \tag{6}$$

where each state variable is calculated by a difference equation. The two first variables of vector $\varpi(k)$ are obtained by (3). The signal *V(k)* is

$$\text{V(k+1)=e(k+1)+V(k); e(k)=r(k)-y(k)} \tag{7}$$

The continuous time reference variables are

$$\begin{bmatrix} \dot{r}(t) \\ \dot{r}_d(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ r_d(t) \end{bmatrix} \quad , \dot{r} = Rr \tag{8}$$

This system generates a sinusoidal reference when started with the values *r(0)=0* and $\dot{r}(0) = \omega V_p$, where V_p is the sine wave amplitude and ω is the angular frequency. In the discrete form, using a sample period T_s, the subsystem (8) is given by

$$n(k+1) = R_d n(k)\ ;\ \ n(k) = [r(k)\ \ r_d(k)]^T\ ,\ R_d = e^{RTs} \tag{9}$$

Therefore, the closed loop system representation becomes

$$\begin{bmatrix} x(k+1) \\ V(k+1) \\ n(k+1) \end{bmatrix} = \begin{bmatrix} Ad & 0 & 0 \\ -C_d A_d & 1 & C_d R_d \\ 0 & 0 & R_d \end{bmatrix} \begin{bmatrix} x(k) \\ V(k) \\ n(k) \end{bmatrix} + \begin{bmatrix} B_d \\ -C_d B_d \\ 0 \end{bmatrix} u_{LQR}(k), \tag{10}$$

$$y(k) = [C_d\ \ 0\ \ \ 0][x(k)\ \ V(k)\ \ \ n(k)]^T$$

The optimal gains of the control law (5) are those that minimize

$$J = \frac{1}{2}\sum_{k=0}^{\infty}\left\{W^{T}(k)QW(k) + u^{T}(k)R_{u}u(k)\right\} \tag{11}$$

where Q and R_u are chosen as positive definite matrices that set the weighting of each state and of the control signal. The K gains can be obtained as

$$K = (H^{T}PH + R_{u})^{-1}H^{T}PG \;;\; P(k+1) = Q + G^{T}P(k)G - G^{T}P(k)H\left[Ru + H^{T}P(K)H\right]^{-1}H^{T}P(k)G \tag{12}$$

3.2 Repetitive Controller (RP)

The RP controller has the duty of improving the steady state response to cyclic disturbances. The block diagram of a system with such controller is sketched in Fig. 3, where c_1 and c_2 are the controller coefficients and $n=f_c/f$ that f and f_c are the reference and sampling frequencies respectively. It was proved that using a RP controller; the steady state error of periodic disturbances with frequency of less than $f_c/2$ is zero [7].

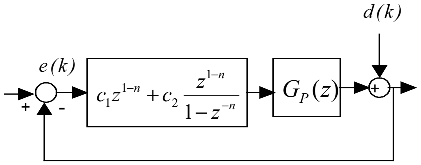


Fig. 3. A discrete time system with a repetitive controller.

The RP control signal can be stated as

$$u_{Rp}(k) = c_1 e(k+1-n) + c_2 \sum_{i=1}^{L} e(k+1-in) \tag{13}$$

In practice, for the digital control systems it should be $k+1-in>0$ so

$$L < \frac{k+1}{n} \tag{14}$$

where L is the number of periods that should be considered.

In [5] it is shown that using c_1 and c_2 in the form of forgetting exponential parameters, the THD can be improved significantly in comparison fixed parameters. The final control signal will be

$$u(k) = u_{LQR}(k) + u_{RP}(k) \tag{15}$$

4 Controller Design Considerations

4.1 Tuning the Controller Gains

To find a proper set of LQR gains for a wide range of loads, the following procedure was adopted:

1. Define the cost function J giving the importance of each state by choice of the matrixes Q and R_u.

2. Obtain the matrices G and H for the specified load condition. A RLS is used to find the plant parameters on line.
3. Find the LQR gains.
4. Verify the system response.
5. If the result has a periodic error, adjust the repetitive gains (using GAs) to correct the system and try again.
6. Else, if the result is not acceptable, change the cost function and repeat the algorithm from 1.

4.2 Recursive Least Square (RLS) Estimator

To estimate the plant parameters when the load conditions are variable, a RLS algorithm is used [6]. The discrete plant model with a zero order hold is given by the following

$$Y(z)=\frac{\theta_3 z+\theta_4}{z^2+\theta_1 z+\theta_2}U(z)+W(z) \tag{16}$$

where $Y, U,$ and W are output, input and disturbance, respectively. In RLS estimation method, a disturbance can affect on the estimated parameters. In this work, for the objective of disturbance attenuation a proper filter, $G_f(z)$, is used. So, the output is

$$y_f(k)=G_f(z)y(k) \quad (17)$$

The difference equation of the estimated output is

$$\hat{y}_f(k)=\left[-y_f(k-1) \quad -y_f(k-2) \quad u_f(k-1) \quad u_f(k-2)\right]\begin{bmatrix}\theta_1(k-1)\\ \theta_2(k-1)\\ \theta_3(k-1)\\ \theta_4(k-1)\end{bmatrix} \tag{18}$$

The RLS equations are

$$\begin{aligned}
&\varphi_f(k)=\left[-y_f(k-1) \quad -y_f(k-2) \quad u_f(k-1) \quad u_f(k-2)\right]\\
&K(k)=p(k-1)\varphi_f(k)(1+\varphi_f{}^T(k)p(k-1)\varphi_f(k))^{-1}\\
&\begin{cases} if \ \left|y_f(k)-\varphi_f{}^T(k)\hat{\theta}(k-1)\right|>\Delta \Rightarrow p(k)=p(0)\\ else \ \ p(k)=(1-K(k)\varphi_f{}^T(k))p(k-1)\end{cases}\\
&\hat{\theta}(k)=\hat{\theta}(k-1)+K(k)(y_f(k)-\varphi_f{}^T(k)\hat{\theta}(k-1))
\end{aligned} \tag{19}$$

where K is the estimation gain matrix, P is the covariance matrix, φ_f is the regression vector and θ is the parameter vector. In (19) the following values are set

$$P(0)=5\times 10^6 \text{ and } \Delta=0.0005\,.$$

And for the filter

$$G_f(z)=\frac{0.002774}{z-0.9972}\cdot$$

After estimation, the state space model of the system is

$$\hat{A}_d=\begin{bmatrix}-\theta_1 & 1\\ -\theta_2 & 0\end{bmatrix},\ \hat{B}_d=\begin{bmatrix}\theta_3\\ \theta_4\end{bmatrix},\ \hat{C}_d=\begin{bmatrix}1 & 0\end{bmatrix} \tag{20}$$

4.3 Observer Design

As only the output voltage is measured, an observer should be designed to estimate the other state variable. In (20) the state variables are the output voltage and *x(k),* where *x(k)* is a function of the output voltage and the inductor current. So, instead of the inductor current, *x(k)* will be estimated. For this, the state space model of the system is rewritten as follows

$$\begin{bmatrix} V_c(k+1) \\ \cdots\cdots\cdots \\ x(k+1) \end{bmatrix} = \begin{bmatrix} A_{d11} & \vdots & A_{d12} \\ \cdots & \cdots & \cdots \\ A_{d21} & \vdots & A_{d22} \end{bmatrix} \begin{bmatrix} V_c(k) \\ \cdots\cdots \\ x(k) \end{bmatrix} + \begin{bmatrix} B_{d11} \\ \cdots\cdots \\ B_{d21} \end{bmatrix} u(k) \text{ , } y(k) = \begin{bmatrix} 1 & : & 0 \end{bmatrix} \begin{bmatrix} V_c(k) \\ x(k) \end{bmatrix} \tag{21}$$

So,

$$\hat{x}(k)=(A_{d22}-k_eA_{d12})\hat{x}(k-1)+k_eV_c(k)+(A_{d21}-k_eA_{d11})V_c(k-1)+(B_{d21}-k_eB_{d11})u(k-1)\text{; } k_e=\frac{A_{d22}+\alpha}{A_{d12}} \tag{22}$$

where α is the desired pole of the observer system and should be faster than the fastest pole of the system. In here it is $\alpha = -0.02$.

4.4 Total Harmonic Distortion Criterion

Total Harmonic Distortion (THD) criterion determines the quality of the output signal, which shows that how much of the signal is not constructed of first harmonic. In order to calculation of THD the coefficients of the Fourier series of the output signal should be firstly obtained. The output is a sinusoidal signal with frequency of *60Hz* and the sampling period $T_s = 1/3600 \sec$. So, the discrete time output is a signal with period of $N = 60$. The coefficients of the Fourier series of the output signal can be obtained as

$$\begin{cases} v_c[n] = \sum\limits_{k=0}^{N/2} c_k e^{jk(2\pi/N)n} \\ c_k = \dfrac{1}{N} \sum\limits_{n=0}^{N-1} v_c[n] e^{-jk(2\pi/N)n} \end{cases} \tag{23}$$

So,

$$\begin{cases} v_c[n] = \sum\limits_{k=0}^{N/2} a_k \sin k(2\pi/N)n + \sum\limits_{k=0}^{N/2} b_k \cos k(2\pi/N)n \\ a_k = \dfrac{2}{N} \sum\limits_{n=0}^{N-1} v_c[n] \sin k(2\pi/N)n \\ b_k = \dfrac{2}{N} \sum_{n=0}^{N-1} v_c[n] \cos k(2\pi/N)n \end{cases} \tag{24}$$

The amplitude of each harmonic can be obtained as the following

$$V_n = \sqrt{a_n^2 + b_n^2} \tag{25}$$

Thus, the THD criteria is calculated by

$$THD = \frac{1}{V_1} \left[\sum_{k=0}^{30} V_k^2 \right]^{1/2} 100\%k \quad ; k \neq 1 \tag{26}$$

where V_k is the amplitude of the k^{th} harmonic.

5 GA Optimization

Evolutionary algorithms are optimisation and search procedures inspired by genetics and the process of natural selection. This form of search evolves throughout generations improving the features of potential solutions by means of biologically inspired operations. On the ground of the structures undergoing optimisation the reproduction strategies, the genetic operators' adopted, evolutionary algorithms can be grouped in: evolutionary programming, evolution strategies, classifier systems, genetic algorithms and genetic programming.

The genetic algorithms behave much like biological genetics [10]. The genetic algorithms are an attractive class of computational models that mimic natural evaluation to solve problems in a wide variety of domains [11,12]. A genetic algorithm comprises a set of individual elements (the population size) and a set of biologically inspired operators defined over the population itself etc. a genetic algorithms manipulate a population of potential solutions to an optimisation (or search) problem and use probabilistic transition rules. According to evolutionary theories, only the most suited elements in a population are likely to survive and generate offspring thus transmitting their biological heredity to new generations [13].

5.1 GA Operations

5.1.1 Selection

The purpose of parent selection in a GA is to give more reproductive changes to those individuals that are the fit. There are many ways to do it, but one commonly used technique is *roulette wheel parent selection* (RWS). A second very popular way of selection is *stochastic universal sampling* (SUS). This way is a single-phase algorithm with minimum spread and zero bias. In this work, we will use SUS.

5.1.2 Crossover (Recombination)

The basic operator for producing new chromosomes in the GA is that of crossover. Like in nature, crossover produces new individuals, which have some parts of both parents' genetic material. The simplest form of crossover is that of single–point crossover [8], which is used in the paper. The crossover probability is set to ρ_c. Of course there exist other crossover variations such as dual point, multipoint, uniform, shuffle, asexual crossover, and single child crossover.

5.1.3 Mutation

Mutation causes the individual genetic representation to be changed according to some probabilistic rule. In the binary string representation, mutation cause a random bit to change its state. In natural evolution, mutation is randomly applied with low probability ρ_m, typically in the range 0.001 and 0.01, and modifies element in the chromosomes. Given that mutation is generally applied uniformly to an entire population of string, it is possible that a given binary string may be mutated at more than one point.

5.1.4 Reinsertion

Once selection and recombination of individuals from the old population have produced a new population, the fitness of the individuals in the new population may be determinate. If fewer individuals are produced by recombination than the size of the original population, than the fractional difference between the new and old population sizes in termed a generation gap. To maintain the size of the original population, the new individuals have to be reinserted into the old population. Similarly, if not all the new individuals are to be used at each generation or if more offspring are generated than size of old population then a reinsertion scheme must be used to determine which individuals are to exist in the new population.

5.2 Fitness Function

In the optimization algorithms, a predefined fitness function should be optimized. As THD is the main objective in our control strategy, it is chosen as a fitness function. So, (33) is the fitness function should be optimized using GA.

6 Simulation Results

For simulation the system parameters shown in Tab. 1 are used. A GAs with population size of 100 and $\rho_c = 1, \rho_m = 0.01$ is chosen as an optimizer. Initial population is chosen from the intervals $0 \le \alpha \le 5, 0 \le \beta \le 0.999$. Fig. 4 depicts the fitness evaluation versus generation number. Evidently, the generation number of 20 can satisfy our objectives.

Tab. 1. System parameters.

Filter inductance	L=5.3 mH
Filter capacitance	*C=80* μ *F*
Nonlinear load	TRIAC in series with *R=24* Ω Turn on angles= 60^o and 240^o
DC input voltage	E=40 V
Reference voltage	$r(t) = 30\sin(120\pi t)$
Sample time	T_s*=1/3600 sec*
State weightings	Q=diag[50,10,150,1,1]
Control weighting	R_u*=100*

The identified parameters using RLS estimation algorithm are shown in Fig. 5. As it can be seen the parameters are estimated excellent. The rapid variations in the parameters are because of sudden load variations. The following parameters are obtained after GAs based optimization

$$\alpha_{opt} = 1.2353, \beta_{opt} = 0.8577 \Rightarrow THD_{opt} = 2.9735$$

Fig. 6 shows the reference and actual outputs produced using the controller and the error between them. Clearly, the output could regulate to its reference value excellent and the error between them is not significant. The harmonic distortion is depicted in Fig. 7.

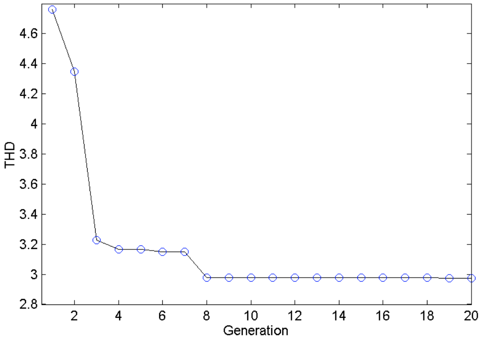


Fig. 4. Fitness evaluation using GAs.

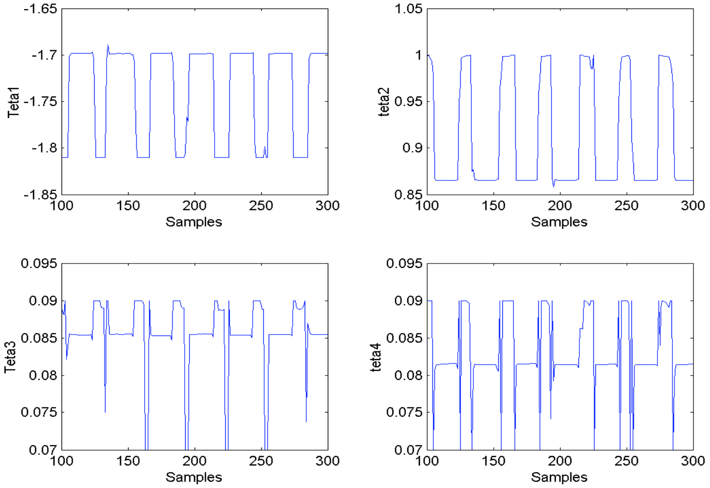


Fig. 5. Identified parameters using RLS estimation algorithm.

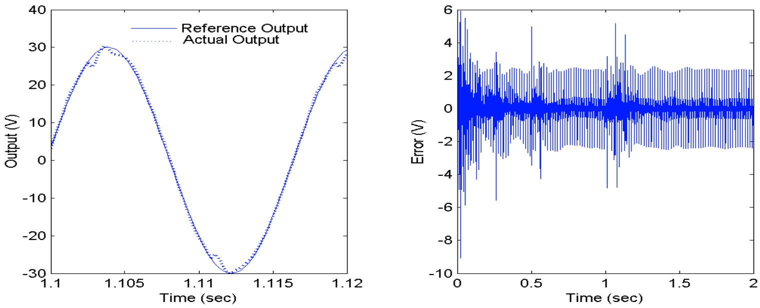


Fig. 6. Reference and actual output with proposed adaptive controller.

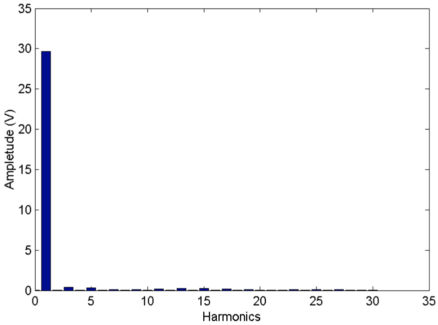


Fig. 7. Harmonic distribution of the output voltage.

7 Conclusion

In this paper, the parameters of RP controller in adaptive LQR-RP controller were tuned based on Gas and the proposed controller was applied to a UPS system. Using RLS estimation method, the system parameters were identified and then the LQR parameters were determined adaptively. The RP controller with exponentially forgetting coefficients was added to the adaptive LQR controller to attenuate the effects of periodic errors. The parameters of these coefficients were tuned using a GAs, which led to an output signal with low THD. Simulation results of the closed-loop system with obtained parameters showed excellent performance of the controller.

References

1. Grundling, Carati H. A. and Pinheriro E.G., A Robust Model Reference Adaptive Controller for UPS Applications, *23rd Int. Conf. on Industrial Appl.,* pp.901-905, 2000.
2. Rashid, M.H, *Power Electronics Circuits, Devices and Applications*, Prentice Hall Inc, 1988.
3. Barnes, L.G. and Krishman, R., An adaptive three-phase UPS inverter controller, *IEEE Power Electronics Specialist Conf.,* pp.473-479, 1995.
4. Montagner, V.F., Carati, E.G. and Grundling, H.A. An Adaptive Linear Quadratic Controller Applied to Uninterruptible Power Supplies, *IEEE Industry Appl. Conf.*, pp.2231-2236, 2000.
5. Shabani, K. and Jalili-Kharaajoo, M., Application of adaptive LQR with repetitive control to uninterruptible power supplies, *2003 IEEE Conference on Control Applications*, Turkey, 2003.
6. Astrom K.J. and Wittenmark V. E., *Adaptive Control. Second edition*, Prentice Hall Inc, 1995.
7. Inoue, T., Practice repetitive control system design, *29th IEEE CDC*, pp.1673-1578, 1990,
8. Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.
9. Haneyoshi T., Kawamura ,A. and Hoft R. G. , Waveform Compensation of PWM Inverter with Cyclic Fluctuating Loads, *IEEE Transaction on Industry Application*, 24(4), pp.874-881, 1998.
10. Grant, K., An Introduction to Genetic algorithms, *CC++ Journal*, 6(2), pp.231-245, 1995.
11. Srinivas, M. and L.M. Patnaik, *Genetic Algorithms: A Survey*, IEEE press, 1994.

12. Filho R.C. and C.P. Treleaven, *Genetic Algorithm Programming Environments*, IEEE press, 1994.
13. Semeraro G.P., Evolutionary Analysis Tools For Real-Time Systems, *sixth Int. symposium on Modeling, analysis and Simulation of Computer and Telecommunication Systems,* Montreal, 1998.
14. Czeczot J., Metzger M., Babary J.P. and Nihtila M., Filtering in Adaptive control of Disturber Parameter Bioreactors in the Presence of Noisy Measurement. *Sim. Prac. The.,* 8, pp.39-56, 2000.
15. Ho, H.L., Rad, A.B., Chan, C.C. and Wong, Y.L., Comparative Studies of Three Adaptive Controllers. *ISA Transactions*, 38, pp.43-53, 1999.

A Comparison of Two Circuit Representations for Evolutionary Digital Circuit Design

Nadia Nedjah and Luiza de Macedo Mourelle

Department of Systems Engineering and Computation, Faculty of Engineering,
State University of Rio de Janeiro,
Rio de Janeiro, Brazil
{nadia, ldmm}@eng.uerj.br
http://www.eng.uerj.br/~ldmm

Abstract. In this paper, we study two different circuit encoding used for digital circuit evolution. The first approach is based on genetic programming, wherein digital circuits consist of their data flow based specifications. In this approach, individuals are internally represented by the abstract trees of the corresponding circuit specifications. In the second approach, digital circuits are thought of as a map of rooted gates. So individuals are represented by two-dimensional arrays of cells. Each of these cells consists of the logic gate name together with the corresponding input signal names. Furthermore, we compare the impact of both individual representations on the evolution process of digital circuits. Evolved circuits should minimise space. We show that for the same input/output behaviour, employing both approaches yield circuits of almost the same characteristics in terms of space. However, the evolution process is much shorter with the second encoding.

1 Introduction

Evolutionary hardware [7] is a hardware that is yield using simulated evolution as an alternative to conventional-based electronic circuit design. Genetic evolution is a process that evolves a set of individuals, which constitutes the population, producing a new population. Here, individuals are hardware designs. The more the design obeys the constraints, the more it is used in the reproduction process. The design constraints could be expressed in terms of hardware area and/or response time requirements. The freshly produced population is yield using some genetic operators such as crossover and mutation that attempt to simulate the natural breeding process in the hope of generating new design that are fitter i.e. respect more the design constraints. Genetic evolution is usually implemented using genetic algorithms.

The problem of interest consists of choosing the best encoding for evolving *rapidly* efficient and creative circuits that implement a given input/output behaviour without much designing effort. The obtained circuits are expected to be minimal both in terms of space and time requirements: The circuits must be compact i.e. use a reduced number of gates and efficient, i.e. produce the output in a short response time. The response time of a circuit depends on the number and the complexity of the

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 594-604, 2004.

gates forming the longest path in it. The complexity of a gate depends solely on the number of its inputs.

The remainder of this paper is divided in five sections. In Section 2, we describe the principles of evolutionary hardware. In Section 3, we describe the first methodology and propose the necessary genetic operators. In Section 4, we describe the second approach as well as the corresponding genetic operators. In Section 5, we compare the evolved hardware using either methodology. Finally, we draw some conclusions.

2 Principles of Evolutionary Hardware Design

Evolutionary hardware consists simply of hardware designs evolved using genetic algorithms, wherein chromosomes represent circuit designs. In general, evolutionary hardware design offers a mechanism to get a computer to provide a design of circuit without being told exactly how to do it. In short, it allows one to automatically create circuits. It does so based on a high level statement of the constraints the yielded circuit must obey to. The input/output behaviour of the expected circuit is generally considered as an omnipresent constraint. Furthermore, the generated circuit should have a minimal size.

Starting form random set of circuit design, which is generally called *initial population*, evolutionary hardware design breeds a population of designs through a series of steps, called *generations*, using the Darwinian principle of natural selection. Circuits are selected based on how much they adhere to the specified constraints. *Fitter* individuals are selected, *recombined* to produce off-springs which in turn should suffer some *mutations*. Such off-springs are then used to populate of the next generation. This process is iterated until a circuit design that obeys to all prescribed constraints is encountered within the current population.

Each circuit within the population is assigned a value, generally called *fitness*. A circuit design is *fit* if and only if it satisfies the imposed input/output behaviour. A circuit design is considered *fitter* than another if and only if it has a smaller size.

An important aspect of evolutionary hardware design is thus to provide a way to evaluate the adherence of evolved circuits to the imposed constraints. These constraints are of two kinds. First of all, the evolved circuit design must obey the input/output behaviour, which is given in a tabular form of expected results given the inputs. This is the truth table of the expected circuit. Second, the circuit must have a reduced size. This constraint allows us to yield compact digital circuits.

We estimate the necessary area for a given circuit using the concept of gate equivalent. This is the basic unit of measure for digital circuit complexity [6]. It is based upon the number of logic gates that should be interconnected to perform the same input/output behaviour. This measure is more accurate that the simple number of gates [6]. The number of gate equivalent and an average propagation delay for each kind of gate are given in Table 1. The data were taken form [6]. Note that only 2-input gates NOT, AND, OR, XOR, NAND, NOR, XNOR and 2:1-MUX are allowed.

Table 1. Gates, symbols and number of gate-equivalent

Name	Symbol	Gate-equivalent	Name	Symbol	Gate-equivalent
NOT		1	NAND		1
AND		2	NOR		1
OR		2	XNOR		3
XOR		3	MUX		3

Let us formalise the fitness function. For this purpose, let C be a digital circuit that uses a subset (or the complete set) of the gates given in Table 1. Let *Gates*(C) be a function that returns the set of all gates of C. On the other hand, let *Val*(T) be the Boolean value that C propagates for the input Boolean vector T assuming that the size of T coincides with the number of input signal required for C. The fitness function is given as follows, wherein X represents the input values of the input signals while Y represents the expected output values of the output signals of C, n denotes the number of output signals that C has.

$$Fitness(C) = \sum_{j=1}^{n} \left(\sum_{i \mid Val(X_i) \neq Y_{i,j}} Penalty \right) + \sum_{g \in Gates(C)} GateEquivalent(g)$$

For instance, consider the evolved circuit described in Equation 2. It should propagate the output signals of Table 2 that appear first (i.e. before symbol /) but it actually propagates the output signals that appear last (i.e. those after symbol /). Observe that signals Z_2 and Z_1 are correct for every possible input combination of the input signals. However, signal Z_0 is correct only for the combinations 1010 and 1111 of the input signals and so for the remaining 14 combinations, Z_0 has a wrong value and so the circuit should be penalised 14 times. Applying function *Gates* to this circuit should return 5 AND gates and 3 OR gates. If penalty is set to 10 then, function *Fitness* should return 140 + 5 × 2 + 3 × 1. This fitness sums up to 153. Note that for a correct circuit the first term in the definition of function *Fitness* is zero and so the value returned by this function is the area of the evaluated circuit.

$$Z_2 \Leftarrow (X_0 \text{ AND } Y_0) \text{ AND } (X_1 \text{ AND } Y_1) \quad (1)$$
$$Z_1 \Leftarrow (X_1 \text{ AND } Y_1) \text{ AND } (X_0 \text{ NAND } Y_1) \quad (2)$$
$$Z_0 \Leftarrow (X_1 \text{ NAND } Y_0) \text{ AND } (X_0 \text{ NAND } Y_1) \quad (3)$$

Apart from fitness evaluation, encoding of individuals is another of the important implementation decisions one has to take in order to use evolutionary computation in general and hardware design in particular. It depends highly on the nature of the problem to be solved. There are several representations that have been used with success: *binary encoding* which is the most common mainly because it was used in the first works on genetic algorithms, represents an individual as a string of bits; *permutation encoding* mainly used in ordering problem, encodes an individual as a sequence of integer; value encoding represents an individual as a sequence of values that are some evaluation of some aspect of the problem; and *tree encoding* represents an individual as tree. Generally, the tree coincides with the concrete tree as opposed to abstract tree of the computer program, considering the grammar of the programming language used. In the next sections, we investigate two different internal representations of digital circuits and look at the pros and cons of both of the methodologies.

Table 2. Truth table of the circuit whose specification is given above in Equation (2)

X_1	X_0	Y_1	Y_0	Z_2	Z_1	Z_0	X_1	X_0	Y_1	Y_0	Z_2	Z_1	Z_0
0	0	0	0	0/0	0/0	0/1	1	0	0	0	0/0	0/0	0/1
0	0	0	1	0/0	0/0	0/1	1	0	0	1	0/0	0/0	1/0
0	0	1	0	0/0	0/0	0/1	1	0	1	0	0/0	1/1	1/1
0	0	1	1	0/0	0/0	0/1	1	0	1	1	0/0	1/1	1/0
0	1	0	0	0/0	0/0	0/1	1	1	0	0	0/0	0/0	0/1
0	1	0	1	0/0	0/0	0/1	1	1	0	1	0/0	0/0	1/0
0	1	1	0	0/0	0/0	1/0	1	1	1	0	0/0	1/1	1/0
0	1	1	1	0/0	0/0	1/0	1	1	1	1	1/1	0/0	0/0

3 Circuit Designs = Programs

In this first approach [4], a circuit design is considered as register transfer level specification. Each instruction in the specification is an output signal assignment. A signal is assigned the result of an expression wherein the operators are those listed in Table 1.

3.1 Encoding

We encode circuit specifications using an array of concrete trees corresponding to its signal assignments. The i^{th} tree represents the evaluation tree of the expression on the left-hand side of the i^{th} signal assignment. Leaf nodes are labelled with a literal representing a single bit of an input signal while the others are labelled with an operand. For instance, a 2-bit multiplier has 4-bit result signal so an evolved register transfer level specification is as follows, wherein the input operands are $X = <x_1x_0>$ and $Y = <y_1y_0>$ and the output is the product $P = <p_3p_2\ p_1p_0>$. A chromosome with respect to this encoding is shown in Fig. 1.

$$p_3 \quad \Leftarrow (x_0 \text{ AND } y_0) \text{ AND } (x_1 \text{ AND } y_1) \tag{4}$$

$$p_2 \quad \Leftarrow (\overline{x}_0 \text{ OR } \overline{y}_0) \text{ AND } (x_1 \text{ AND } y_1) \tag{5}$$

$$p_1 \quad \Leftarrow (\overline{x}_1 \text{ OR } \overline{y}_0) \text{ XOR } (\overline{x}_0 \text{ OR } \overline{y}_1) \tag{6}$$

$$p_0 \quad \Leftarrow (y_0 \text{ AND } x_0) \text{ OR } y_0 \tag{7}$$

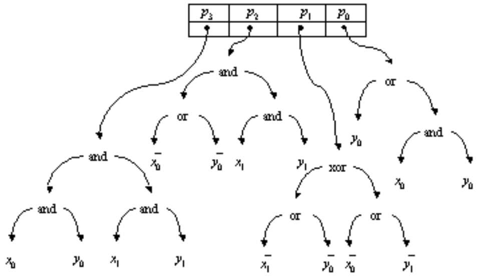


Fig. 1. Chromosome for the circuit specification of Equations (4) to Equation (7) with respect to the first encoding

3.2 Genetic Operators

Crossover recombines two randomly selected individuals into two fresh off-springs. It may be *single-point*, *double-point* or *uniform* crossover [3]. Crossover of circuit specification is implemented using a variable double-point crossover [3] as described in Fig. 2.

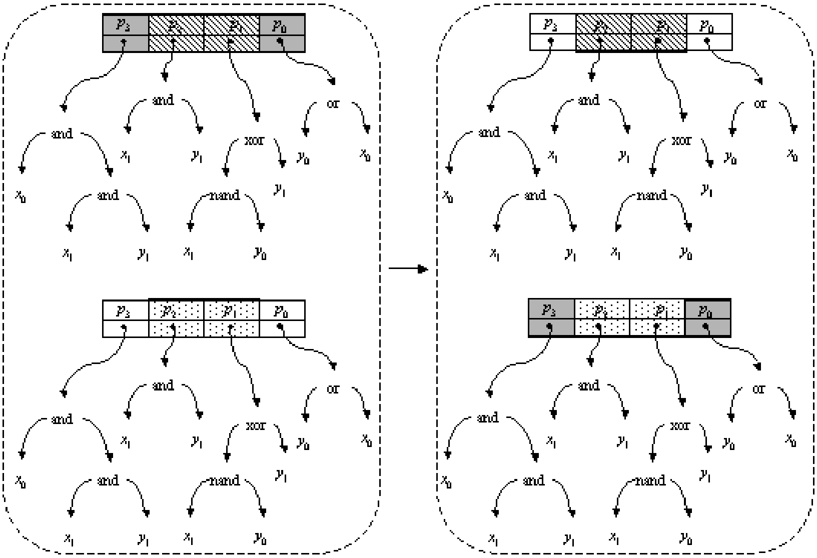
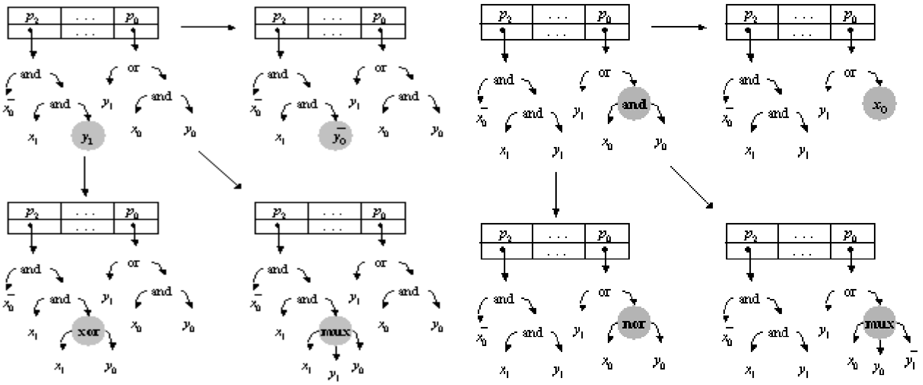


Fig. 2. Double-point crossover of circuit specification

One of the important and complicated operators for genetic programming [2] is the *mutation*. It consists of changing a gene of a selected individual. The number of individuals that should suffer mutation is defined by the *mutation rate* while how many genes should be altered within an individual is given by the *mutation degree*.



(a) Operand node mutation ***(b) Operator node mutation***

Fig. 3. Node mutation for circuit specification

Here, a gene is the expression tree on the left hand side of a signal assignment symbol. Altering an expression can be done in two different ways depending the node that was randomised and so must be mutated. A node represents either an operand or operator. In the former case, the operand, which is a bit in the input signal, is substituted with either another input signal or *simple* expression that includes a single operator as depicted in Fig. 3-(a). The decision is random. In the case of mutating an operand node to an operator node, we proceed as Fig. 3-(b).

4 Circuit Designs = Schematics

Instead of textual specification, a circuit design can also be represented by a graphical one, which is nothing but the corresponding schematics. So in this second approach [5], a circuit design is considered as map of gates given in Table 1. The first row of gates receives the input signals while the last row delivers the circuit output signals.

4.1 Encoding

We encode circuit schematics using a matrix of cells that may be interconnected. A cell may or may not be involved in the circuit schematics. A cell consists of two inputs or three in the case of a MUX, a logical gate and a single output. A cell may draw its input signals from the output signals of gates of previous rows. The gates includes in the first row draw their inputs from the circuit global input signal or their complements. The circuit global output signals are the output signals of the gates in the last raw of the matrix. A chromosome with respect to this encoding is given in Fig. 5.

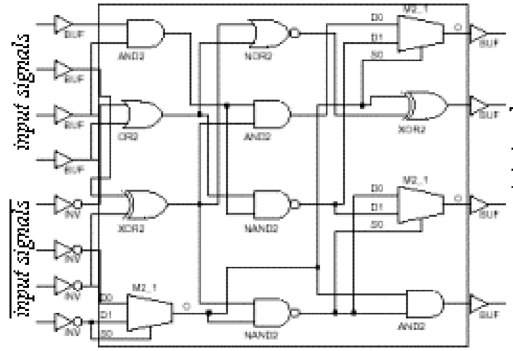


Fig. 5. Chromosome with respect to the second encoding

4.2 Genetic Operators

Crossover of circuit schematics, as for specification crossover, is implemented using a variable four-point crossover. This is described in Fig. 6.

The mutation operator can act on two different levels: gate mutation or route mutation. In the first case, a cell is randomised and the corresponding gate changed. When a 2-input gate is altered by another 2-input gate, the mutation is thus completed. However, when a 2-input gate is changed to a 3-input gate (i.e. to a MUX), the mutation process randomises an additional signal among those allowed (i.e. all the input signals, their complements and all the output signals of the cells in the rows previous). Finally, when a MUX is mutated to a 2-input gate, the selection signal is simply eliminated. The second case, which consists of route mutation is quite simple. As before, a cell is randomised and one of its input signals is chosen randomly and mutated using another signal. The mutation process is described in Fig. 7.

5 Result Comparison

In this section, we compare the evolutionary circuits yield when using the first encoding and those evolved using the second encoding. For this purpose, we use five examples that were first used in [1]. For each one of these examples, we provide the expected input/output behaviour of the circuit and the circuit designs evolved by the first and the second approach. For each of the evolution process, we state the number of generation for the same input parameters.

The truth table of the benchmarks is given in Table 3. Apart from benchmark *(a)*, which has only three inputs, the remaining benchmarks, i.e. *(b)*, *(c)* and *(d)*, need four inputs. On the other hand, apart from benchmark *(d)*, which yields three output signals, the rest of the benchmarks, i.e. *(a)*, *(b)* and *(c)* propagate a single output signal. The circuits receive input signal *X* and propagate output signal *Y*.

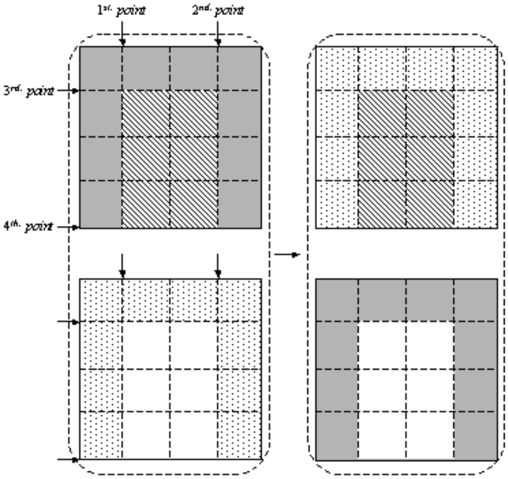


Fig. 6. Four-point crossover of circuit schematics

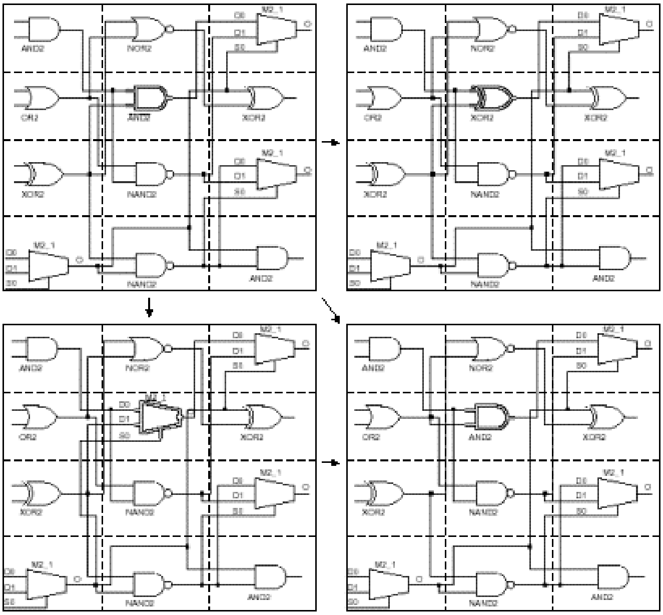


Fig. 7. Mutation of circuit schematics

Table 3. Truth table of the circuit used as benchmarks to compare both encoding methods

X_3	X_2	X_1	X_0	Y_a	Y_b	Y_c	Y_{d_2}	Y_{d_1}	Y_{d_0}
0	0	0	0	0	1	1	1	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	0	1	0	1	0
0	0	1	1	1	1	0	0	1	0
0	1	0	0	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	0
0	1	1	0	1	1	1	0	1	0
0	1	1	1	0	1	1	0	1	0
1	0	0	0	-	1	1	0	0	1
1	0	0	1	-	0	1	0	0	1
1	1	1	0	-	1	1	1	0	0
1	0	1	1	-	0	0	0	1	0
1	0	0	0	-	0	0	0	0	1
1	1	0	1	-	1	1	0	0	1
1	1	1	0	-	0	1	0	0	1
1	1	1	1	-	0	1	1	0	0

The circuit evolved by the genetic algorithm that uses the first encoding are shown by the corresponding schematics in Fig. 8, while those discovered by the genetic algorithm that uses the second encoding are depicted also by the corresponding schematics in Fig. 9.

In Table 4, we provide the size of the circuits evolved in terms of gate-equivalent as well as the average numbers of generations that were necessary to reach the shown result. The last size corresponds to human designs. These numbers of generations were obtained for 100 different evolutions. In the last column of Table 4 we computed the average evolution effort gain in terms of percentage. The average gain over 400 different evolution processes is of about 87% in favour of the second encoding.

6 Conclusions

In this paper, we described two different encoding to be used in digital circuit evolution. In the first representation, we view a circuit design as a program that provide its specification. The second representation, in contrast, views a circuit as schematics that graphically describe the composition of the circuit. For each one of these encoding, we presented the necessary genetic operators. Finally, we compared the performance of two genetic algorithms, each one using one of the described encoding. The comparison was performed using 4 different benchmarks that were first used in [1]. The evolution process was repeated 100 times for each one of the benchmarks and the average number of generation that was required to reach a valid solution was recorded. The evolved circuits are of the same complexity in terms of size. However, the number of generation necessary to reach a valid solution when using the first encoding was almost 10 times bigger. We computed the average gain in terms of generation number. This was about 87% in favour of the second encoding. The evolved solution are always better compared the ones classically designed.

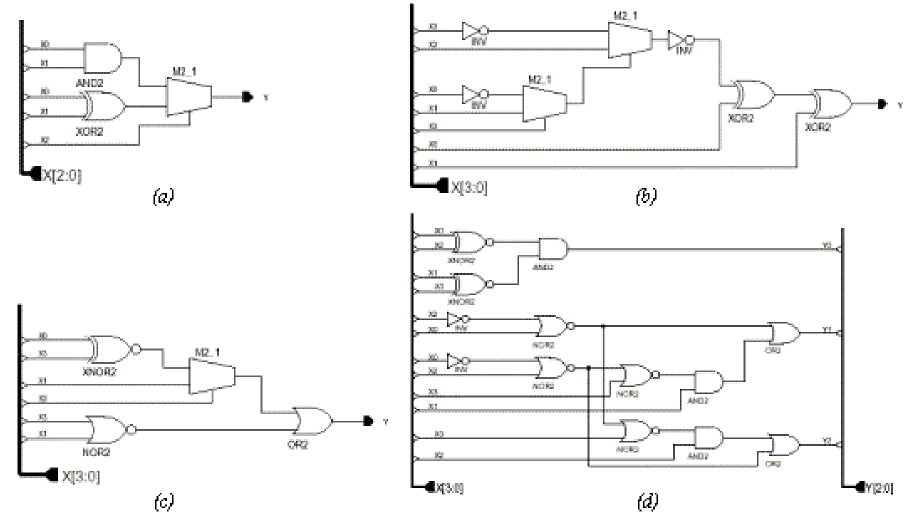


Fig. 8. Circuits evolved when using the first encoding for the benchmarks whose input/output behaviour is given in Table 3

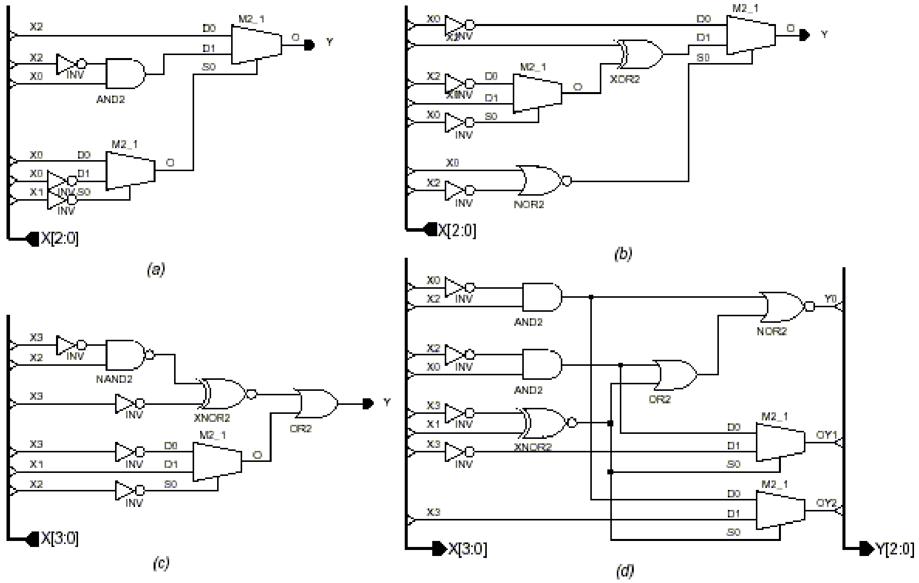


Fig. 9. Circuits evolved when using the second encoding for the benchmarks whose input/output behaviour is given in Table 3

Table 4. Number of gate-equivalents and generations that were necessary to evolve the circuits of Fig. 8 and 9 when using the first and the second encoding respectively

Benchmark	**Size$_1$**	**Size$_2$**	**Size$_3$**	**#Gen$_1$**	**#Gen$_2$**	**gain**
(a)	8	8	12	302	29	90%
(b)	13	10	20	915	72	92%
(c)	9	9	20	989	102	89%
(d)	20	16	34	1267	275	78%

References

1. Coelho, A.A.C., Christiansen, A.D. and Aguirre, A.H., Towards Automated Evolutionary Design of Combinational Circuits, Computation and Electrical Eng., 27, pp. 1-28, (2001)
2. Koza, J. R., Genetic Programming. MIT Press, (1992)
3. Miller, J.F., Thompson, P. and Fogarty, T.C., Designing Electronics Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study, In Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, Wiley Publisher (1997)
4. Nedjah, N. and Mourelle, L.M., Evolvable Hardware Using Genetic Programming, Proc. 4th International Conference Intelligent Data Engineering and Automated Learning - IDEAL 2003, Hong Kong, LNCS, Vol. 2690, Springer-Verlag, (2003)
5. Poli, R. Efficient evolution of parallel binary multipliers and continuous symbolic regression expressions with sub-machine code GP, Technical Report CSRP-9819, University of Birmingham, (1998)
6. Rhyne, V.T., Fundamentals of digital systems design, F.F. Kuo (Ed.) Prentice-Hall Electrical Engineering Series, (1973)
7. Thompson, A., Layzel, P. and Zebelum, R.S., Explorations in design space: unconventional design through artificial evolution, IEEE Transactions on Evolutionary Computations, 3(3):81-85, (1996)

Motif-Oriented Representation of Sequences for a Host-Based Intrusion Detection System

Gaurav Tandon, Debasis Mitra, and Philip K. Chan

Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, USA
{gtandon, dmitra, pkc}@cs.fit.edu

Abstract. Audit sequences have been used effectively to study process behaviors and build host-based intrusion detection models. Most sequence-based techniques make use of a pre-defined window size for scanning the sequences to model process behavior. In this paper, we propose two methods for extracting variable length patterns from audit sequences that avoid the necessity of such a pre-determined parameter. We also present a technique for abstract representation of the sequences, based on the empirically determined variable length patterns within the audit sequence, and explore the usage of such representation for detecting anomalies in sequences. Our methodology for anomaly detection takes two factors into account: the presence of individual malicious motifs, and the spatial relationships between the motifs that are present in a sequence. Thus, our method subsumes most of the past works, which primarily based on only the first factor. The preliminary experimental observations appear to be quite encouraging.

1 Introduction

In this work we propose a technique for intrusion detection from audit logs. Audit logs of a hosting computer store process activities at the system call level, which are typically analyzed to study the behavior. Various approaches have been proposed to detect intrusions based upon identifying deviations from normal traffic at a host computer. Forrest et al. (1996) proposed an approach in which they formed correlations between audit events within a fixed window size. Lane and Brodley (1997a, 1997b) suggested techniques for capturing the user profiles. They calculated the degree of similarity between two different audit sequences by looking into adjacent events within a window of ten events. The main limitation of such approaches is that an attack may be spread beyond such a pre-determined window.

In this paper, we attempt to address the problem (of using a fixed window-size) by extracting variable length patterns (subsequences or motifs) from the audit sequences. We then transform the audit event-sequence to a pattern-based representation. This also results in some data reduction and is expected to simplify the task of the "similarity" finding algorithm. Another motivation behind developing this representation scheme is not only to detect intrusions but also to model different types of normal or abnormal behaviors. Moreover, most of the anomaly detection systems

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 605-615, 2004.

rely on the presence/absence of some events that represent the abnormal behavior. But an attack may not always be such a novel event. Rather an attack may constitute normal events placed in a "wrong" order. We conjecture that the relative positions of some frequently observed subsequences (we call them *patterns, sub-strings* or *motifs* synonymously) within a sequence of system calls model the behavior of a sequence.

The paper is organized as follows. In Section 2, we describe some system call sequence-based approaches in IDS's. Section 3 presents our approach for detecting anomalies, and the steps involved in our experiments. In Section 4, we summarize the results obtained from the experiments with some real data. We also analyze the results and discuss certain related issues there. A brief description of the 1999 DARPA evaluation data set is also presented. Section 5 addresses some views and the future direction of our work.

2 Related Work

Lane and Brodley (1997a, 1997b) examined sequences of user actions and explored various matching functions to compare the behaviors of sequences with the various user profiles. Their matching functions measured similarity of adjacent audit events within a fixed size-window. They empirically found the window-length 10 to be providing better results. Forrest et al. (1996) also proposed an approach for host based anomaly detection wherein the traces of normal executions were considered and the subsequences were recorded by creating a look-ahead window. These subsequences were then used to create a database. All subsequences in the test cases (using the same window size and look-ahead) were checked for consistency with the training database. If the number of mismatches was above a threshold for a sequence it was deemed anomalous. One of the issues involved in their approach was the use of a small window that does not correlate well over a long period of time. Similar sequences with minor variations could still be flagged as anomalous producing a very high number of false positives. Both these techniques have arbitrarily fixed the window length that could lead to an easy evasion by an attacker. Our technique detects *all* frequently occurring subsequences of variable lengths, thereby removing the need for such a pre-defined parameter for window length.

Wespi et al. (1999, 2000) proposed a scheme for producing variable length patterns using Teiresias, a pattern discovery algorithm for biological sequences (Rigoutsos and Floratos, 1998). This method improved upon the fixed length pattern usage techniques proposed by Forrest et al. Jiang et al. (2002) extended this idea by taking into account both the intra-pattern and the inter-pattern anomalies for detecting intrusions. Though all these techniques also use the notion of variable length sequences, they rely on encountering novel events to identify anomalous behaviors. We attempt to find anomalies by using an abstract representation of an audit trail sequence based on the previously known subsequences (we have extracted them from the sequences). Michael (2003) also suggested a technique for replacing frequent subsequences with "meta-symbols" when extracting variable length sequences. But his approach looks for repeated substrings that are only adjacent to one another. Our suggested technique is a generalization of his approach.

A simple methodology for comparing amino acid sequences or nucleotide sequences, called the *Matrix Method*, was proposed by Gibbs and McIntyre (1970). A matrix is formed between two sequences, where the elements of one sequence constitute the rows and that for the other sequence constitute the columns. An entry, where the row and the column have common elements between them, is marked with an asterisk (*), as shown in Figure 1. A series of adjacent asterisks parallel to the diagonal indicate a common subsequence between the two sequences. We have borrowed this technique for extracting common motifs of system calls. Another popular method for alignment in bio-informatics is the Basic Local Alignment Search Tool (BLAST, Altschul et al, 1990). This algorithm starts with small common subsequence as a seed and builds it gradually until mismatches appear. There is a whole body of literature in the area of bio-informatics on alignment of bio-sequences. However, they are not exactly useful for our problem of motif generation for three reasons. (1) These algorithms are for aligning two (or more) sequences and not for extracting common sub-sequences. (2) Point mutation through evolution creates gaps or replaces an element with another and the alignment algorithms need to accommodate for that. (3) One of the objectives of such alignment algorithms is fast database-search over some sequence databases and not necessarily for identifying motifs. The primary goal there is to create a distance metric between bio-sequences so that one could retrieve matched sequences against a query sequence. There are many details of the problem structure (e.g., typically a small query against the large target sequences) that are not necessarily valid in our case. However, this body of literature provides a good source for our research.

3 Approach

Filtering the audit data: In this preprocessing phase we extract the very large finite length sequences of system calls from the audit trails, related to a particular application (e.g., *ftpd*) and related to a particular process.

Translation of system calls: Subsequently we map each system call to a unique arbitrary symbol. Thus, we form strings of finite lengths from the sequences of system calls.

3.1 Motif Extraction Phase

We have pursued two different mechanisms for generating the motifs: *auto-match* and *cross-match*. These two techniques are discussed below.

3.1.1 Motif Extraction Using Auto-match

For our initial experiments, we have considered any *pattern* (*substring/motif*) occurring more than once in a string as frequent enough to be a candidate motif. We start by looking at such sub-strings of length two within each string. Then, we do the same for substrings of increasing lengths (3, 4, … up to an arbitrary limit of 7).

Variable length motifs are generated by studying the overlapping patterns of the fixed length motifs in the sequences (explained latter). The longer motifs may be generated out of the motifs with shorter length. While the frequencies of occurrence of the motifs within a sequence provide good indication regarding such subsumption, there are some problems associated with the issue. To illustrate this consider the following sequence

acggcggfgjcggfgjxyz (I)

One may note that in this sequence we have a motif *cgg* with frequency 3, and another motif *cggf* with frequency 2, which is longer and sometimes subsumes the shorter motif but not always. We consider them as two different motifs since the frequency of the shorter motif was more than the longer one. Thus, a motif of shorter length may be subsumed by a longer motif only if it has the same frequency as that of the shorter motif. Frequency of a shorter motif in a sequence cannot obviously be less than that of the longer one.

Another aspect of the above issue is that two or more overlapping motifs may be merged together to form a motif of greater length. This is how we create motifs of arbitrarily long sizes. After extracting motifs of length 4 in the example sequence *(I)*, we have motifs *cggf, ggfg* and *gfgj*, all with frequency 2. Since these patterns are overlapping and have the same frequency, they may be merged together in order to obtain a longer motif *cggfgj* with the same frequency 2. However, there are instances when the smaller motifs may concatenate forming a longer motif at some places, but may occur at other positions on the sequences independently (i.e., not overlapping). Even though they have the same frequency, the concatenated longer motif does not subsume the shorter ones. Consider the sequence c*ggfgjabcggfpqrggfgxyzgfgj*. Here, the motifs *cggf, ggfg* and *gfgj* each have a frequency 2. But the longer motif *cggfgj* occurs only once, though we may wrongly conclude a frequency of 2 by using the above derivation. The remaining instances of the smaller motifs are at different (non-overlapping) positions within the string. The solution to this problem is that the occurrence of the longer motif obtained from the fusion of the smaller motifs should be verified for accuracy. If the frequency of the longer motif is found to be the same as that of a smaller one, then merging of the latter ones is all right, i.e., we ignore the smaller motifs. This technique reduces the effort of going through all possible string lengths and of finding all possible motifs for those lengths. The procedure of finding motifs of variable lengths by first merging and then verifying is repeated until no more motifs could be merged.

Let us consider the following two synthetic sequences: *acgfgjcgfgjxyzcg* (II), *cgfgjpqrxyzpqr* (III). Using *auto-match* for sequence *(II)*, we obtain the motifs *cg* with frequency 3, *gf* with frequency 2, and *cgf*, *cgfg* and *cgfgj* each with frequency 2. Motifs *cgf* and *cgfg* are subsumed by *cgfgj*, so the final list of motifs for sequence *(II)* is *cg* and *cgfgi*. For sequence *(III)*, the only motif we get by auto-match is *pqr*.

After the procedure terminates running over a string the motifs extracted are (1) of length ≥ 2, (2) a motif is a substring of another motif iff the former exists independently, and (3) the frequency of each motif is ≥ 2. Motifs extracted from each string are added to the motif database making sure that there is no redundancy in the latter, i.e., the same motif appearing in different strings is not recorded twice. In subsequent phases of modeling sequence-behaviors, the frequency values of the motifs are no longer needed.

3.1.2 Motif Extraction Using Cross-Match

The technique of auto-match is meant for finding frequent patterns in each string. Some other techniques (Jiang et al, 2002; Michael, 2003; Wespi et al, 1999, 2000) also follow the concept of extracting such frequent patterns in each string. However, we are additionally interested in patterns that did not occur frequently in any particular string but are present across more than one of the different strings. We believe that these motifs could also be instrumental in modeling an intrusion detection system, as they might be able to detect attack patterns across different attack sequences. We obtained these by performing pairwise *cross-match* between different sequences. Using cross-match between the example sequences *(II)* and *(III)*, we get the motifs *cgfgj* and *xyz*, since these are the maximal common subsequences within the two given sequences, as shown in Figure 1, which represents the two sequences in a matrix and depicts the common elements with a * (Gibbs and McIntyre, 1970).

	Sequence (II)																
		a	c	g	f	g	j	c	g	f	g	j	x	y	z	c	g
Sequence (III)	c		*														
	g			*													
	f				*												
	g					*											
	j						*										
	p																
	q																
	r																
	x												*				
	y													*			
	z														*		
	p																
	q																
	r																

Fig. 1. Matrix representation of the cross-match strategy between the sequences II &III

Motifs extracted in cross-match are (1) of length ≥ 2, (2) appears at least in a pair of sequences. Unique motifs extracted in this phase are also added to the database. Note that the motifs extracted in the cross-match phase could be already present (and so, ignored) in the motif database generated in the previous phase of auto-match. Also note that, although a motif could be appearing only once across a pair of sequences in the cross-match, its frequency count is 2 because of its presence in both the matched sequences.

A discerning point between our work and most of the previous ones in the IDS-literature is that we do not necessarily attach any semantics (e.g., anomalous motif) to any sub-string, although we could do so if that is necessary. We extract motifs that are just “frequently” appearing and so, may have some necessity within the sequences, and we use them only for creating an abstract representation of the sequences (explained later). Since the motif database is created only once the efficiency of the procedures for extracting them is not a critical issue.

3.2 Sequence Coverage by the Motif Database

A motif database storing all the extracted unique motifs is created from the previous experiments. Continuing with the example sequences *(II)* and *(III)*, the motif database comprises of (obtained by auto-match and cross-match) *cg*, *cgfgj*, *pqr* and *xyz*.

We also produce data reduction by our proposed representation of sequences using only the relevant motifs. Hence, it is important to verify the degree of loss of information. After extracting the motifs, we performed another study of checking the coverage of the sequences by these motifs. *Coverage* of a sequence by the motif-database is parameterized by the percentage of the sequence length that coincides with some motifs in the database. For instance, the string *acgfgjcgfgjxyzcg* has 15 characters from the motif database {*cg*, *cgfgj*, *pqr* and *xyz*} out of total length of 16, so, has coverage of 93.8%. The technique proposed by Wespi et al (1999, 2000) determines anomaly when the coverage is below a threshold. But in reality an uncovered pattern may not necessarily be a representation of an attack. Rather, it may be a novel audit sequence representing a normal behavior. This increases the rate of false alarms in their method. We do not use coverage information explicitly; rather we use it only for the purpose of checking the quality of motif extraction process. It is good to have a significant amount of coverage of the sequences by the motif database, which will indicate a low loss of information for the representation.

3.3 Motif-Based Representation of Sequences

At first (matching phase) we extract the motifs’ presence in each sequence. A standard automaton is being implemented for all the motifs to find the respective ones within a sequence in this phase. Each sequence is run through the automaton and the existing motifs along with their starting positions in the sequence are recorded.

Eventually each motif in the database is translated to a unique numeral id. [It is important to note that in the pre-processing phase the translation was for individual system calls to the corresponding ids, while the translation here is for the motifs.] Each string is subsequently represented as a scatter-plot, where a motif’s starting point is the abscissa and the motif’s id is the ordinate of the corresponding point. This is how the spatial relationships between different relevant subsequences/motifs within a sequence are being taken into account. The representations for the two synthetic example sequences *(II)* and *(III)* are shown in Figures 2(a) and 2(b) respectively.

3.4 Testing Phase – Anomaly Detection

Let $T=\{t_1,t_2,...,t_N\}$ be the training set comprising of N sequences and the test sequence be denoted by *s*. Also, let $M=\{m_{1,}m_2,...,m_n\}$ denote the set of *n* motifs. $f(m_i,t_k)$ represents the frequency of motif m_i in the sequence t_k., whereas $d_{m_i}(t_k)$ is the distance between two consecutive occurrences of the same motif m_i in the sequence t_k. The re-occurrence interval of a motif m_i in the sequence t_k , denoted by $r_{m_i}(t_k)$, is computed by averaging over all such motif distances within the sequence. This is represented as

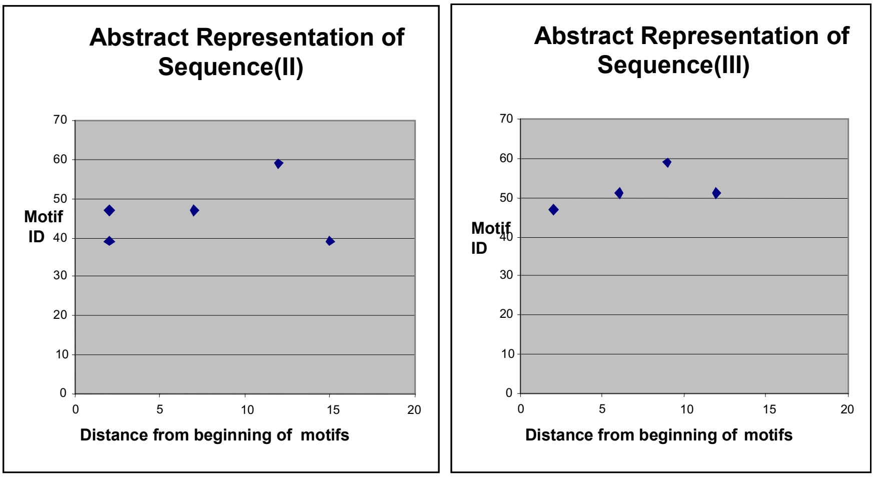
$$\forall m_i \in s,t_k \;\&\; f(m_i,s), f(m_i,t_k) \geq 2, \quad r_{m_i}(t_k) = \frac{\sum_{t_k} d_{m_i}(t_k)}{f(m_i,t_k)-1}$$

An anomaly score $AS(m_i,s,t_k)$ is associated with every motif m_i common between a pair of sequences *s* and t_k, which is equal to the difference of the re-occurrence intervals for that motif in the two sequences. The total anomaly score $TAS(s,t_k)$ is obtained by aggregating the anomaly scores over all such motifs.

$$TAS(s,t_k) = \sum_{m_i \in s,t_k} AS(m_i,s,t_k) = \sum_{m_i \in s,t_k} | r_{m_i}(t_k) - r_{m_i}(s) |$$

An alarm is raised if the anomaly score for a test sequence with respect to each of the training sequences exceeds the threshold. That is,

$$\forall k \in [1,N], \quad \textit{Total Anomaly Score}(s,t_k) > \textit{Threshold} \quad \Rightarrow \quad \textit{Anomalous Sequence}$$



(a) (b)

Fig. 2. Cloud representations of the sequences *acgfgjcgfgjxyzcg* and *cgfgjpqrxyzpqr* respectively

4 Results And Analysis

The data set that we have used for the extraction of motifs is developed by Lippman et al. (2000) for the 1999 DARPA Intrusion Detection Evaluation. The test bed involved a simulation of an air force base that had some machines under frequent attack. These machines comprised of Linux, SunOS, Sun Solaris and Windows NT. Various IDS's are being evaluated using this test bed, which comprised of three weeks of training data obtained from the network sniffers, audit logs, nightly file system dumps and BSM logs from Solaris machine that trace the system calls. We selected the Solaris host and used the BSM audit log (Osser and Noordergraaf, 2001) to form the sequences for the processes corresponding to the *ftpd* application. For a given process id, all the data from the *exec* system call to the *exit* system call comprised the data for that particular process. The *fork* system call was handled in a special manner. All the system calls for a child process are for the same application as the parent process until it encounters its own *exec* system call. Thus, we have used variable length sequences corresponding to different processes running over the ftp server. We have selected a total of 91 sequences for the *ftpd* application across weeks 3, 4 and 5 of the BSM audit log. In our experiments, the sequence lengths were mostly in a range of 200-400, with an average sequence length of 290.11 characters.

Table 1. Coverage's of the sequences by motifs

BSM Data Used	Range of Coverage (Auto-Match)		Range of Coverage (Cross-Match)	
	Lowest	Highest	Lowest	Highest
Week 3	82 %	94 %	91 %	99 %
Week 4	65 %	94 %	90 %	99 %
Week 5	65 %	92 %	96 %	99 %

The first set of experiments involved motif extraction using auto-match and cross-match methods, as explained before. We generated 66 unique motifs in the auto-match and a total of 101 unique motifs after the cross-match, for our motif database. The next set of experiments computed the percentage coverage of the audit sequences by the motifs to check the quality of the motif database for the studied sequences. Table 1 presents the lowest and the highest coverage over the sequences. It shows better coverage for cross-match since it considers motifs across sequences ignoring their frequencies. A good coverage of the sequence by the motifs implies here that there will be a low loss of information in our scatter-plot representation.

After obtaining the motifs, we created the abstract representations of the sequences as described in Section 3.3. They are the scatter-plots for the sequences in the *ftpd* application of the 1999 DARPA evaluation data set. Our next set of experiments aimed at generating alarms based upon the anomalies observed, as discussed in Section 3.4. Week 3 comprised of the training sequences and weeks 4 and 5 were used as the test data as they contained unlabeled attacks. The evaluation criterion is the number of true positives and false positives obtained by using our methodology. Utilizing a simple metric for anomaly score (average separation of each motif's multiple occurrences in a sequence) as explained in Section 3.4 we are able to detect 6 attacks with 3 false alarms. We were able to successfully detect 1 warezmaster, 2 warezclient, 2 ftp-write and 5 instances of a guessftp attack. Whereas warezmaster and warezclient attacks are denial-of-service attacks, ftp-write and guessftp belong to the remote-to-local attack category. More sophisticated metric will be deployed in future for automated clustering of the sequences.

Figure 3 depicts curves for two test sequences – a normal test sequence and an attack test sequence respectively. Each point on a curve corresponds to the anomaly score (Y-axis) of the corresponding test sequence with respect to the training sequences (each with a unique id as on X-axis). The results indicate that the execution of a normal process results in less deviation of relative motif positions and hence a lower anomaly score. A malicious sequence, on the other hand, consists of execution of code which alters the spatial relationship between the motifs, resulting in anomaly scores (with respect to all training sequences) exceeding the threshold. We chose a threshold of 100 for our experiments. We also varied the threshold and raised it till 300 but there was no change in the results.

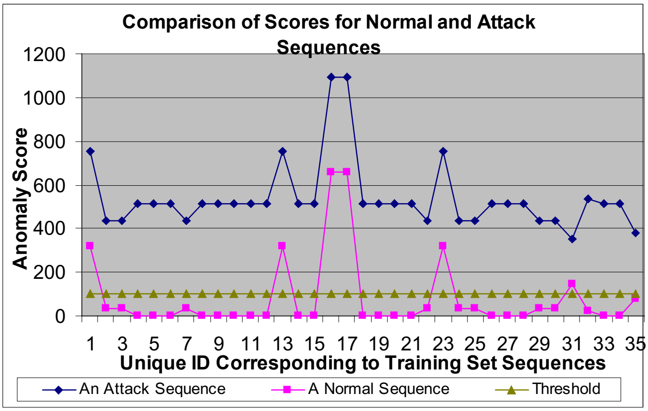


Fig. 3. Anomaly score curves for a normal and an attack sequence (from the *ftpd* application in the BSM audit log of the 1999 DARPA evaluation data set). Each point on a curve represents the anomaly score of the sequence with respect to some sequence (denoted by unique sequence ids on the X-axis) from the training set

We would also like to mention that an attacker might devise some clever techniques to evade typical sequence-based anomaly detection systems. Wagner and Soto (2002) presented one such idea wherein they were successful in modeling a malicious sequence by adding null operators to make it consistent with the sequence of system calls. The sequence based techniques dealing with short sub-string patterns can be bypassed by spreading the attack over longer duration (or longer sub-sequences). Our technique uses variable length motifs and also takes the relative positions of the motifs for anomaly detection, and hence will be robust against such evasion. In essence, our system models sequences at two different levels – at the individual motif level and also at the level of spatial relationship between motifs within the audit sequence. The latter level adds to the security of the system and would make it even harder for the attacker to evade the system, since he has to now not only use the "normal" audit event patterns, but also to place those event-sequences/motifs within the respective sequence at proper relative positions.

5 Conclusions and Future Work

We have described two different lines of approach for extracting motifs from finite length strings. We were successful in generating frequently occurring motifs using the first technique (auto-match) and motifs that were common (but not necessarily frequent) across different sequences using the cross-match technique. Motifs obtained from the cross-match technique displayed better coverage across all the sequences. These pattern-extracting techniques can also be used for data reduction without much loss of information, since we now deal with only the subsequences (motifs) and not the whole audit trail. The structure of a sequence is accurately captured by the motifs and a good coverage across the sequence implies low inherent loss of information.

We have also presented an abstract motif-based representation of sequences that preserves the relative positions between the motifs in a sequence. This abstract representation enhances the efficiency and effectiveness of the resulting intrusion detection system. Our results suggest that not only are the novel system call sequences important, but also the relative positions of the motifs/patterns are important in modeling the activities over a host-system. Of course, if an attack occurs with an event or a motif not existing in our motif database we will not able to detect it. However, our model can be extended to incorporate such cases.

We intend to use an unsupervised learning technique in future, which will not only be able to distinguish the activities as normal or abnormal but also will be able to classify a richer repertoire of system behaviors. Such knowledge of a particular attacker's behavior may also help in tracking him (or them) down.

Acknowledgement. National Science Foundation has partly supported (IIS-0296042) this work.

References

1. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic Local Alignment Search Tool. *Jnl. Of Molecular Biology*, vol. 215, pp. 403-410.
2. Forrest S., Hofmeyr S., Somayaji A., and Longstaff T. (1996). A Sense of Self for UNIX Processes. In *Proceedings of the1996 IEEE Symposium on Research in Security and Privacy*, pages 120-128. IEEE Computer Society, IEEE Computer Society Press.
3. Gibbs A.J. and McIntyre G.A. (1970). The diagram, a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem.* 16:1-11.
4. Jiang N., Hua K., and Sheu S. (2002). Considering Both Intra-pattern and Inter-pattern Anomalies in Intrusion Detection. In *Proceedings ICDM.*
5. Lane, T., and Brodley, C.E. (1997a). Detecting the abnormal: Machine Learning in Computer Security, (TR-ECE 97-1), West Lafayette, IN: Purdue University.
6. Lane, T., and Brodley, C.E. (1997b). Sequence Matching and Learning in Anomaly Detection for Computer Security. In *Proceedings of AI Approaches to Fraud Detection and Risk Management.*
7. Lippmann R., Haines J., Fried D., Korba J., and Das K. (2000). The 1999 DARPA Off-Line Intrusion Detection Evaluation. *Computer Networks* (34) 579-595.
8. Michael, C.C. (2003). Finding the vocabulary of program behavior data for anomaly detection. *Proc. DISCEX'03.*

9. Osser W., and Noordergraaf A. (February 2001). Auditing in the Solaris™ 8 Operating Environment. Sun Blueprints™ Online.
10. Rigoutsos, Isidore and Floratos, Aris. (1998). Combinatorial pattern discovery in biological sequences. *Bioinformatics*, 14(1):55–67.
11. Wagner D., Soto P. (2002). Mimicry Attacks on Host-Based Intrusion Detection Systems. *ACM Conference on Computer and Communications Security*.
12. Wespi A., Dacier M., and Debar H. (1999). An Intrusion-Detection System Based on the Teiresias Pattern-Discovery Algorithm. *Proc. EICAR*.
13. Wespi A., Dacier M., and Debar H. (2000). Intrusion detection using variable-length audit trail patterns. In *Proceedings of RAID 2000, Workshop on Recent Advances in Intrusion Detection*.

Computational Intelligent Techniques for Detecting Denial of Service Attacks

Srinivas Mukkamala[1] and Andrew H. Sung[1,2]

[1] Department of Computer Science
[2] Institute for Complex Additive Systems Analysis
New Mexico Tech
Socorro, New Mexico 87801, U.S.A.
{srinivas|sung}@cs.nmt.edu

Abstract. Recent cyber attacks on the Internet; have proven that none of the open network systems are immune to availability attacks. Recent trend of the adversaries "if I can't have it, nobody can" has changed the emphasis on the information resource availability in regards to information assurance. Cyber attacks that are launched to deny the information resources, data and services to the legitimate user are termed as denial of service attacks (DoS) or availability attacks. The distributed nature of the Internet helps the adversary to accomplish a multiplicative effect of the attack; such attacks are called distributed denial of service attacks.

Detecting and responding to denial of service attacks in real time has become an elusive goal owing to the limited information available from the network connections. This paper presents a comparative study of using support vector machines (SVMs), multivariate adaptive regression splines (MARS) and linear genetic programs (LGPs) for detecting denial of service attacks. We investigate and compare the performance of detecting DoS based on the mentioned techniques, with respect to a well-known sub set of intrusion evaluation data gathered by Lincoln Labs.

The key idea is to train the above mentioned techniques using already discovered patterns (signatures) that represent DoS attacks. We demonstrate that highly efficient and accurate signature based classifiers can be constructed by using computational intelligent techniques to detect DoS attacks. Future we describe our ongoing effort of using computational intelligent agents to respond to DoS attacks at the boundary controllers of the network.

1 Introduction

Information availability has become an issue of serious global concern. The complexity and openness of client/server technology combined with Internet have brought about great benefits to the modern society; meanwhile, the rapidly increasing connectivity and accessibility to the Internet has posed a tremendous security threat. Malicious usage, attacks, and sabotage have been on the rise as more and more computers are put into use. Connecting information systems to networks such as the Internet and public telephone systems further magnifies the potential for exposure through a variety of attack channels. These attacks take advantage of the flaws or

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 616-624, 2004.

omissions that exist within the various information systems and software that run on many hosts in the network.

In the early days security of information was equated to confidentiality. With the change in adversaries trend that denying legitimate services to authorized users is as good as compromising the confidentiality of information; information availability has gained additional emphasis.

Efforts on how to define and characterize denial of service attacks through a collection of different perspectives such as bandwidth, process information, system information, user information and IP address is being proposed by several researchers [1,2,3]. Using the defined characteristics a few signature-based and anomaly based detection techniques are proposed [4,5,6,7,8,9]. Recent distributed availability attacks proved that there exists no effective means to detect, respond and mitigate availability attacks.

In this paper we propose a computational intelligent approach to detect denial of service attacks. A comparative study of SVMs, MARs and LGPs for detecting denial of service attacks is performed through a variety of experiments performed on a well know Lincoln Labs data set that consists of more than 80% of different denial of service attacks described in section 3. We also address the issue of identifying the best performing computational intelligent technique in three critical aspects, Accuracy, training time and testing time to build signature-based detection models for detecting availability attacks.

In the rest of the paper, a brief introduction to the data we used is given in section 2. In section 3 a brief introduction to DoS attacks is given. Experiments for detecting DoS attacks using MARs, SVMs and LGPs are given in section 4. Conclusions of our work are presented in section 6.

2 Data

In the 1998 DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks [10,11]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [12]. Of this database a subset of 494021 data were used, of which 20% represent normal patterns.

Attack types fall into four main categories:

1. Probing: surveillance and other probing
2. DoS: denial of service
3. U2Su: unauthorized access to local super user (root) privileges
4. R2L: unauthorized access from a remote machine

3 Denial of Service Attacks

Attacks designed to make a host or network incapable of providing normal services are known as denial of service attacks. There are different types of DoS attacks: a few of them abuse the computers legitimate features; a few target the implementations bugs; and a few exploit the misconfigurations. DoS attacks are classified based on the services that an adversary makes unavailable to legitimate users. A few examples include preventing legitimate network traffic, preventing access to services for a group or individuals.

Table 1. Overview of denial of service attacks

Attack Type	Service	Mechanism	Effect of the attack
Apache2	http	Abuse	Crashes httpd
Back	http	Abuse/Bug	Slows down server response
Land	http	Bug	Freezes the machine
Mail bomb	N/A	Abuse	Annoyance
SYN Flood	TCP	Abuse	Denies service on one or more ports
Ping of Death	Icmp	Bug	None
Process table	TCP	Abuse	Denies new processes
Smurf	Icmp	Abuse	Slows down the network
Syslogd	Syslog	Bug	Kills the Syslogd
Teardrop	N/A	Bug	Reboots the machine
Udpstrom	Echo/Chargen	Abuse	Slows down the network

Attack signatures:

- *Apache2*: DoS attack performed against an apache web server where an adversary submits an http request with several http headers. In theory if the server receives too many of such requests it will slow down the functionality of the web server and eventually crashes. This attack denies the web service temporarily; the service can be regained automatically with system administrator intervention.

- *Back*: DoS attack performed against an apache web server where an adversary submits an URL request with several front slashes. While trying to process these requests, the server's service becomes unavailable for legitimate users. This attack denies the web service temporarily; the service can be regained automatically.

- *Land*: DoS attack performed against TCP/IP implementations where an adversary sends a spoofed SYN packet where the source and destination IP address are the same. In theory it's not possible to have the same destination address as the source address. The adversary targets the badly configured networks and uses the innocent machines as zombies for performing distributed attacks. This attack can be prevented by carefully configuring the network, which prevents requests containing the same source and destination IP addresses.

- *Mailbomb*: DoS attack performed against the server where an adversary floods the mail queue, possibly causing failure. The adversary tries to send thousands of mails to a single user. This attack denies the service permanently. The service can be regained by the system administrator intervention; blocking the mails coming from or to the same user within a short period of time can prevent the attack.

- *SYN Flood (*Neptune*):* DoS attack performed against every TCP/IP implementations where an adversary utilizes the half open TCP connections to flood the data structure of half open connections on the innocent server causing to deny access to legitimate requests. This attack in some cases can cause permanent failure. The service can be regained automatically. Looking for a number of simultaneous SYN packets coming form the same host or unreachable host in a given short period of time can prevent this attack.

- *Ping of Death (PoD*): DoS attack performed against older versions of operating systems where an adversary tries to send an oversized IP packet, and the system reacts in an unpredictable manner, causing crashing, rebooting and even freezing in some cases. This attack causes temporary failure of services. Looking for Internet Control Message Protocol (ICMP) packets that are longer than 64000 bytes and blocking them is the way to prevent this attack.

- *Process table*: DoS attack performed against a variety of different Unix systems where an adversary tries to allocate a new process for every incoming TCP/IP connection; when the systems process table is filled completely, legitimate commands are prevented from being executed. This attack causes temporary failure of services. Looking for large number of active connections on a single port helps in preventing this attack.

- *Smurf:* DoS attack performed against all the systems connected to the Internet where an adversary uses the ICMP echo request packets to IP broadcast addresses from remote locations to deny services. This attack causes temporary denial of services and can be automatically recovered. Looking for a large number of echo replies to the innocent machine from different places without any echo request made by the innocent machine helps in detecting this attack.

- *Syslogd*: DoS attack performed against Solaris servers where an adversary tries to kill the syslogd service remotely. The adversary exploits the DNS lookup feature, if the source IP address does match the DNS record then the syslogd crashes with a segmentation fault. This attack permanently denies the services and can be recovered with the system administrator intervention.

- *Teardrop*: DoS attack performed against older versions of TCP/IP stack where an adversary exploits the feature of IP fragment reassembly. This attack denies the services temporally.

- *Udpstrom:* DoS attack performed against networks where an adversary utilizes the UDP service feature to cause congestion and slowdown. This attack denies the services permanently and can be resumed with system administrator intervention. This attack can be identified by looking for spoofed packets and inside network traffic.

4 Experiments

We partition the data into the two classes of “Normal” and “DoS” patterns, where the DoS attack is a collection of six different attacks (back, neptune, ping of death, land, smurf, and teardrop). The objective is to separate normal and DoS patterns. We apply SVMs to the DARPA data set as described in Section 2. In our experiments we use the SVMs, MARs, and LGPs to classify patterns in several different ways. In the first set of experiments SVMs, MARS and LGPs are used to classify normal patterns vs. DoS patterns. In the second set of experiments the above mentioned techniques are used to classify DoS patterns vs. the rest of the patterns, which include other types of attacks. Further we extended our experiments for DoS instance-specific classifications. Results of SVM, MARS and LGP classifications are given in Table 2, Table 3 and Table 4 respectively.

4.1 Support Vector Machines

The SVM approach transforms data into a feature space *F* that usually has a huge dimension. It is interesting to note that SVM generalization depends on the geometrical characteristics of the training data, not on the dimensions of the input space [13,14]. Training a support vector machine (SVM) leads to a quadratic optimization problem with bound constraints and one linear equality constraint. Vapnik shows how training a SVM for the pattern recognition problem leads to the following quadratic optimization problem [15].

Minimize: $$W(\alpha) = -\sum_{i=1}^{l}\alpha_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \quad (4)$$

Subject to $$\sum_{i=1}^{l} y_i \alpha_i \quad (5)$$

$$\forall i : 0 \le \alpha_i \le C$$

Where *l* is the number of training examples α is a vector of *l* variables and each component α_i corresponds to a training example *(x_i, y_i)*. The solution of (4) is the vector α^* for which (4) is minimized and (5) is fulfilled.

Because SVMs are only capable of binary classifications, we will need to employ individual SVMs, for DoS classification and 6- DoS instance classification problem in DoS detection, respectively. (Note: In the DARPA data, the 4 attack classes refine into 32 instances of attacks; however, training data are available for only 23 instances. Classification results using SVMs for DoS detection are given in Table 2.

Table 2. Performance of SVMs for DoS attacks

Experiment	Accuracy (%)	Training time (sec)	Testing time (sec)	Train/Test Data sets
DoS/Normal	99.69	24.09	15.30	11593/***51875***
DoS/Rest	99.25	22.87	1.92	5092/***6890***
Smurf/Rest	100.00	4.79	2.45	5092/***6890***
Neptune/Rest	99.96	21.18	0.87	5092/***6890***
Back/Rest	99.70	8.34	2.87	5092/***6890***
Land/Rest	99.93	0.82	0.15	5092/***6890***
PoD/Rest	99.99	3.18	1.75	5092/***6890***
Teardrop/Rest	99.61	16.16	0.07	5092/***6890***

4.2 Multivariate Adaptive Regression Splines

Splines can be considered as an innovative mathematical process for complicated curve drawings and function approximation. To develop a spline the X-axis is broken into a convenient number of regions. The boundary between regions is also known as a knot. With a sufficiently large number of knots virtually any shape can be well approximated. While it is easy to draw a spline in 2-dimensions by keying on knot locations (approximating using linear, quadratic or cubic polynomial etc.), manipulating the mathematics in higher dimensions is best accomplished using basis functions. The MARS model is a regression model using basis functions as predictors in place of the original data. The basis function transform makes it possible to selectively blank out certain regions of a variable by making them zero, and allows MARS to focus on specific sub-regions of the data. It excels at finding optimal variable transformations and interactions, and the complex data structure that often hides in high-dimensional data [16,17].

Given the number of records in most data sets, it is infeasible to approximate the function $y=f(x)$ by summarizing y in each distinct region of x. For some variables, two regions may not be enough to track the specifics of the function. If the relationship of y to some x's is different in 3 or 4 regions, for example, the number of regions requiring examination is even larger than 34 billion with only 35 variables. Given that the number of regions cannot be specified a priori, specifying too few regions in advance can have serious implications for the final model. A solution is needed that accomplishes the following two criteria:

- Judicious selection of which regions to look at and their boundaries
- Judicious determination of how many intervals are needed for each variable

Given these two criteria, a successful method will essentially need to be adaptive to the characteristics of the data. Such a solution will probably ignore quite a few variables (affecting variable selection) and will take into account only a few variables at a time (also reducing the number of regions). Even if the method selects 30 variables for the model, it will not look at all 30 simultaneously. Such simplification is accomplished by a decision tree at a single node, only ancestor splits are being considered; thus, at a depth of six levels in the tree, only six variables are being used to define the node.

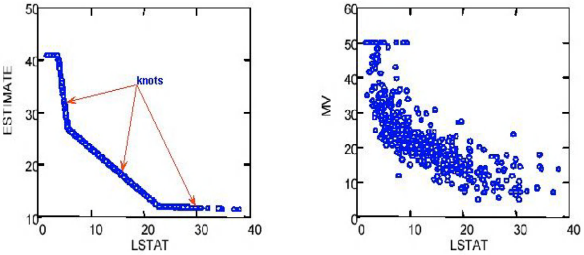


Fig. 1. MARAS data estimation using spines and knots (actual data on the right)

Table 3. Performance of MARS for DoS attacks

Experiment	Accuracy (%)	Train/Test Data sets
DoS/Rest	94.73	5092/*6890*
Smurf/Rest	100.00	5092/*6890*
Neptune/Rest	98.48	5092/*6890*
Back/Rest	99.80	5092/*6890*
Land/Rest	100	5092/*6890*
PoD/Rest	99.90	5092/*6890*
Teardrop/Rest	99.94	5092/*6890*

4.3 Linear Genetic Programming

Linear genetic programming is a variant of the GP technique that acts on *linear genomes* [18,19]. Its main characteristics in comparison to tree-based GP lies in that the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like c/c ++). An alternate approach is to evolve a computer program at the machine code level, using lower level representations for the individuals. This can tremendously hasten up the evolution process as, no matter how an individual is initially represented, finally it always has to be represented as a piece of machine code, as fitness evaluation requires physical execution of the individuals. The basic unit of evolution here is a native machine code instruction that runs on the floating-point processor unit (FPU). Since different instructions may have different sizes, here instructions are clubbed up together to form *instruction blocks* of 32 bits each. The *instruction blocks* hold one or more native machine code instructions, depending on the sizes of the instructions. A crossover point can occur only between instructions and is prohibited from occurring within an instruction. However the mutation operation does not have any such restriction.

Table 4. Performance of LGPs for DoS attacks

Class/ Rest	Population Size	Cross Over Rate	Mutation Rate	Accuracy (%)
DoS	2048	72.24	78.14	99.91
Smurf	512	30.10	96.61	100
Neptune	512	30.10	96.61	99.99
Back	1024	67.97	85.03	99.87
Land	512	40.09	87.45	100
PoD	1024	57.17	87.36	99.91
Teardrop	1024	57.17	87.36	100

5 Conclusions

We have implemented MARs, SVMs and LGPs for detecting DoS patterns and validate their performance using the DARPA intrusion evaluation data. A comparative study on the performance of the computationally intelligent techniques for detecting DoS attacks is presented and LGPs performed the best with the expense of training and testing times.

In the IDS application (and, specifically, DoS detection) SVMs perform well and outperform other machine learning techniques like neural networks in the important respects of scalability, training time, running time and detection accuracy [20,21]. In particular, the training time of SVMs is frequently an order of magnitude shorter than that of neural networks, while the detection accuracy is markedly higher.

SVMs and LGPs easily achieve very high detection accuracy (greater than 99%) for each of the attack instances of data. SVM feature ranking method in our previous work revealed that using only the important or using important plus secondary features achieved only slightly lower accuracy, which suggests that a sensitivity selector may be included in an IDS--i.e., depending on the security requirements, different sets of features may be used in the DoS detection engine.

Acknowledgements. Support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech) and a U.S. Department of Defense IASP capacity building grant is gratefully acknowledged.

References

1. T. Draelos, et. Al, "Distributed Denial of Service Characterization," *Technical Report*, Sandia National Laboratories, 2003.
2. C. Shields, "What do we mean by network denial of service?" *Proceedings of the 2002 IEEE workshop on Information Assurance*. US Military Academy, 2002, pp. 196-203.
3. W. J. Blackert, D. C. Furnanage, and Y. A. Koukoulas, "Analysis of Denial of service attacks Using An address Resolution Protocol Attack, " *Proceedings of the 2002 IEEE Workshop on Information Assurance*. US Military Academy, 2002, pp. 17-22.

4. D. W. Gresty, Q. Shi, and M. Merabti, “Requirements for a general framework for response to distributed denial of service,” *Seventeenth Annual Computer Security Applications Conference*, 2001, pp. 422-229.
5. J. Mirkovic, J. Martin, and P. Reiher, “A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms,” *Technical Report # 020017*, Department of Computer Science, UCLA, 2002.
6. K. Park, and H. Lee, “On the Effectiveness of Router-Based Packet Filtering for Distributed DoS attack and Prevention in Power-Law Internets,” *Proceedings of the SGICOMM’01*, 2001, pp. 15-26.
7. S. Mohiuddin, S. Hershkop, R. Bhan, S. Stolfo, “Defeating Against Large Scale Denial of Service Attack,” *Proceedings of the 2002 IEEE Workshop on Information Assurance*. US Military Academy, 2002, pp. 139-146.
8. S. Mukkamala, and A. H. Sung, “Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligence Techniques,” In International Journal on Digital Evidence, IJDE Volume 1, Issue 4.
9. S. Mukkamala, and A. H. Sung, “Detecting Denial of Service Attacks Using Support Vector Machines,” *Proceedings of IEEE International Conference on Fuzzy Systems*, IEEE Computer Society Press, 2003, pp. 1231-1236.
10. K. Kendall, “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems,” *Master's Thesis, Massachusetts Institute of Technology*, 1998.
11. S. E. Webster, “The Development and Analysis of Intrusion Detection Algorithms,” *S.M. Thesis, Massachusetts Institute of Technology*, 1998.
12. W. Lee, S. J. Stolfo, and K. Mok, “Mining Audit Data to Build Intrusion Detection Models,” *Proceedings of the KDD-98*, honorable mention best application paper, 1998.
13. T. Joachims, “Estimating the Generalization Performance of a SVM Efficiently,” *Proceedings of the International Conference on Machine Learning*, Morgan Kaufman, 2000.
14. T. Joachims, “SVMlight is an Implementation of Support Vector Machines (SVMs) in C,” http://ais.gmd.de/~thorsten/svm_light. University of Dortmund. *Collaborative Research Center on Complexity Reduction in Multivariate Data* (SFB475), 2000.
15. V. M. Vladimir, “The Nature of Statistical Learning Theory,” *Springer*, 1995.
16. J. H. Friedman, Multivariate Adaptive Regression Splines, Annals of Statistics, Vol 19, 1991, pp. 1-141.
17. D. Steinberg, P. L. Colla, and K. Martin, “MARS User Guide,” San Diego, CA: Salford Systems, 1999.
18. W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, “Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications,” *Morgan Kaufmann Publishers, Inc*, 1998.
19. AIMLearning Technology, http://www.aimlearning.com.
20. S. Mukkamala, and A. H. Sung, “Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines,” To appear in Journal of the Transportation Research Board (of the National Academies).
21. S. Mukkamala, G. Janoski, and A. H. Sung, “Intrusion Detection Using Neural Networks and Support Vector Machines,” *Proceedings of IEEE International Joint Conference on Neural Networks*, 2002, pp.1702-1707.

Writer Identification Forensic System Based on Support Vector Machines with Connected Components

Marino Tapiador, Jaime Gómez, and Juan A. Sigüenza

Universidad Autónoma de Madrid,
Computer Engineering Department, Spain
{marino.tapiador, jaime.gomez, jalberto.siguenza}@ii.uam.es

Abstract. Automatic writer identification systems have several applications for police corps in order to perform criminal and terrorist identification. The objective is to identify individuals by their off-line manuscripts, using different features. Character level features are currently the best choice for performance, but this kind of biometric data needs human support to make correct character segmentation. This work presents a new system based on using Connected Component level features, which are close to character level and can be easily obtained automatically. Our experiments use Support Vector Machines using Connected Components gradient vectors to identify individuals with good results. A real-life database was used with real forensic cases in different writing conditions.

1 Introduction

Forensic techniques have been broadly used by police corps in order to perform criminal and terrorist identification. One of these methods is handwriting identification related to forensic methods. Forensic scientists can identify individuals in big populations or previously registered suspects using typographical analysis.

Forensic scientists have developed different handwriting individuality analysis techniques in order to verify or identify criminals using multiple manuscript features. Forensic document examiners have been intensively studying the methodological aspects of handwriting identification, as described in [1] and [2].

The identification of authors of questioned handwritten documents has great importance on the criminal justice system from two points of view: verification and identification. The system described in this paper was focused on identification procedures and it was developed in collaboration with a Spanish police corps forensic laboratory. The identifying problem compares 1-document (questioned) versus N-documents (genuine) i.e. the case of classifying one handwritten document between several authors in a group. For instance, this approach would try to identify a criminal or individual between a group of suspects, supposing there are handwritten samples (specimens) previously recorded from them.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 625-632, 2004.

The field of dynamic biometrics ([3],[4]) is related to the use of some individual's behavior performing at repetitive tasks in order to identify/verify him/her. Off-line handwriting identification can be included as a particular kind of behavioral biometrics where the shapes and relationships of writing strokes are used as biometric features to authenticate an individual. These features are used in the case of forensic handwriting analysis to identify a person in a group of suspects.

Several works related to off-line writer identification can be found in biometrics on previous work [5]-[10], and the most important study about handwriting individuality (due to the great volume of samples used: 1000 volunteers) that we are aware of was performed by Srihari et al. [5]. An important result of this work was to establish that character level features give the best performance for writer identification. But using character features, human support is needed to do manual character segmentation because there aren't perfect character segmentation algorithms. The main objective of our work was to design an automatic system able to perform writer identification with the high identification accuracy of character features but without human interaction i.e., it does not need the human forensic support.

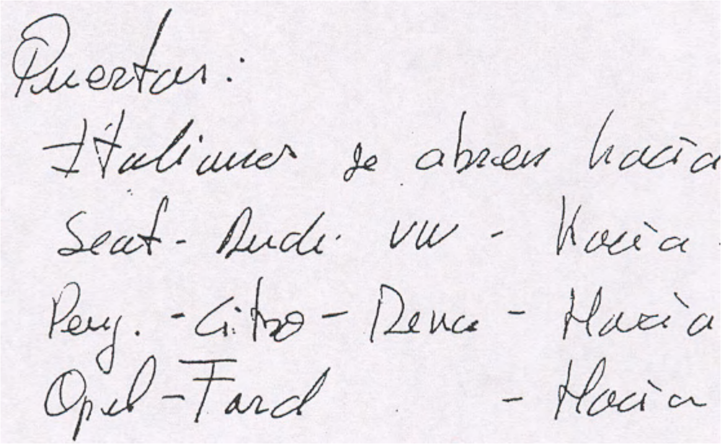
2 Method

The application described in this paper is organized in two main modules: the data capture module, performing the writing specimens digitalization and segmentation, i.e. Connected Components; and the identification module, developed using a Support Vector Machine algorithm.

2.1 Data Capture Module: Connected Components

The first step in the data capture process is the digitization of the handwriting images. The system uses digitized images of the original confiscated/authenticated documents or with photo-copies of them (A4 size), provided by the Spanish forensic lab collaborating in the project. An example of this kind of document is shown by Fig. 1, in this case an extract obtained from a notebook. The documents are scanned into digital form using a resolution of 300 DPI in a PC running Microsoft Windows XP. The full document image is stored in the system's hard disk in bitmap image format (BMP) using 24 bits per pixel i.e. RGB color.

Next step is feature extraction. This subsystem extracts the information needed in the classification subsystem with the defined constraints. As previously mentioned, other works on writer identification have shown that character level features give the best performance, but they are also very difficult to implement because there are no perfect segmentation algorithms at the character level. Character segmentation needs also to solve the handwriting recognition problem, which nowadays is pretty far to be achieved. Therefore, an operator (maybe the forensic expert) has to do manual character segmentation using common graphics tools with the document bitmap image. This is a painfully slow and hard process.



Puertas:
Italianos se abren hacia
Seat - Audi · VW - Hacia
Peug. - Citro - Renau - Hacia
Opel - Ford - Hacia

Fig. 1. Digitized photo-copy of a confiscated document (questioned specimen).

The main objective of our system is to avoid this problem. The system has to be completely automatic and keep a high identification rate. In order to achieve this goal, our tool uses a pseudo-character level: Connected Components. Connected Components are usually used in handwriting recognition in order to get chaincodes (as described by [11],[12]), i.e. a representation of the stroke direction that sometimes is the input for character segmentation algorithms. This algorithm binarizes bitmap image using the Otsu's algorithm [13]. Afterwards, the black and white image is analyzed row by row looking for black pixels in letter/word strokes, and when a black pixel is found the algorithm follows the outside contour of the stroke until it gets back to the initial pixel. This contour defines the Connected Component i.e. a handwriting particle delimited by background (white pixels). In general this is not a letter or letter group, it also can be just an isolated stroke of a letter or just image noise. The Fig.2 illustrates an example of this process where five components are extracted from the Spanish word "Puertas" ("doors") in the document.

The Connected Component level can be good for identification considering the physiology of writing and the factors influencing letter formation. A writer learns to write by the "impulse method" (see [14]). The different writing impulses are defined as follows:

- The stroke impulse: the writer learns to write by drawing individual lines, which, when connected together, make a letter.
- The letter impulse: the whole letter is created as a single writing act.
- The syllable impulse: syllables or several letters are connected together.
- The word or name impulse: complete words or names are created as a single act of writing.
- The sentence/phrase impulse: when "graphic maturity" is reached, the thinking and writing of sentences/phrases are a single act.

These impulse levels are related to regularity levels in the behavior of the writer. This regularity can be exploded by pattern recognition algorithms in order to identify the writer. The Connected Components are graphic specimens separated by background pixels so it means the writer did a pause between them during writing, so they were writing impulses closed to the "syllable impulse level" previously cited. Our system uses this "Connected Component impulse level" for writer identification.

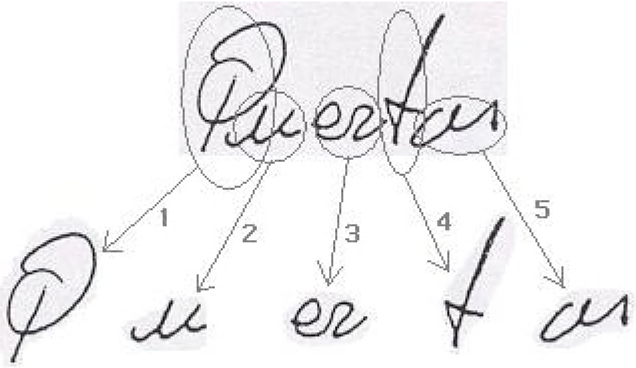


Fig. 2. Connected Components extraction process.

Therefore the tool is able to process the bitmap documents, binarize them, and extract all its Connected Components. All these Connected Component samples are stored in a database. The automatic tool has been developed with Microsoft Visual C++ and it uses a simple Microsoft Access database. The main entities in this database are users and samples. Users are individuals, and a number of different data are registered related to them. One of these data is the 'type' of user. The type of user can be 'questioned' or 'genuine'. A 'genuine' user is a virtual individual in the database who has associated several samples that come from an authenticated/validated individual. The 'questioned' user has associated specimens from a suspect, i.e. he/she is an individual who has to be identified between the group of 'genuine' registered users in the system's database.

Because the forensic procedure we were attempting to automate was based just on Connected Component examination, which is very closed to character examination, we included in the system the best character level feature used in the study of [5]: gradient feature vectors.

The computation of gradient feature vectors is based on the gradient of the letter image. Let be (x,y) a point in the digitized image with a f(x,y) image value, then the gradient vector at this point is a vector where the x and y components are approximated using Sobel spatial filters. These operators consist of 3 x 3 masks that are convoluted in the pixel and the eight nearest neighbors. The gradient vector direction (a value between 0 and 2π radians) is calculated and further approximated to one of twelve ideal directions, having π/6 radians steps. A 13th direction is used in order to denote background points i.e. when all the pixels for convoluting the mask are white (background) the pixel value is set to the 13th direction.

Considering the idea of comparing the stroke shapes of the character by the gradient vector, the algorithm has to make use of relative distances to ignore the effects of character size. The gradient vector is computed cell by cell in a 4 x 4 grid, centered in the image. A gradient vector results with 13 bits (directions) by the 4 x 4 cells i.e. a total vector of 208 bits. The direction bits of each cell are set to 1 when the probability of the corresponding vector is greater than a threshold equal to 1/13, otherwise they are set to 0.

2.2 Identification Module: SVM

Support Vector Machine (SVM) is a classification tool introduced by V. Vapnik in 1995 [15]. Since then, it has been used in a variety of problems with excellent results [16]. A SVM separates two classes using an optimum discriminator hyperplane so to minimize a convex quadratic objective function. This simple algorithm has remarkable properties: there is one solution only (no local minima); SVM parameters are few and easy to handle; data separation is performed in a very high dimensional feature space, making it much easier; new features are not calculated explicitly, so there is no complexity increase regardless of the use of high dimensional spaces; expected noise figures are introduced in the training algorithm, upgrading robustness; generalization capability is outstanding, despite the high dimensional space.

As SVM are binary classifiers, the easiest way to build a multiple class SVM classifier is to create a number of independent one-against-all binary SVM [17]. In the experiments we had 10 genuine individuals, so we created 10 binary SVM, each of them classifying one individual's handwritten characters against the other nine's. Note that a binary equilibrium is essential for good performance, i.e., input data set size must have a similar number of positive and negative examples. For that purpose we increased the influence of positive examples (single individual) to be the same as the sum of negative examples (the other individuals). In this process no information was lost. Each data vector contained the gradient features information only, described as 208 ordered integer values either 1 or 0, as described in previous sections, which correspond to the directions of the gradient in the Connected Component.

Note that when we want to recognize between two writers, their letters and/or Connected Components must be compared from an equivalence point of view, i.e, we should try to compare a's with a's to perform an expert-like process. To achieve implicit character recognition, the best choice is the Gaussian Radial Basis Function as the SVM kernel. We also used parameter sigma as 5 and noise parameter C as 15.

The rate calculation is performed on the questioned individuals. For each one, all the trained SVM are fed with the complete single questioned individual data set. Each SVM gives an independent *proximity* rate: number of positive-classified examples versus number of examples. The SVM with the highest score should then correspond to the genuine individual associated to the questioned set. Otherwise a classification error is generated.

3 Experimental Results

Using the techniques described by previous sections, a real-life database has been created using Spanish documents confiscated to criminals/terrorists or authenticated in the presence of a policeman. A volume of 20,864 Connected Components samples have been captured and digitized, evenly distributed in 20 individuals, 10 genuine and 10 questioned.

The specimens from genuine individuals were obtained from documents written in the presence of a policeman, and the specimens from questioned individuals were obtained from documents confiscated to these same individuals, written at a different time and conditions. All of them correspond to real cases where forensic document examiners have identified them. These data was provided by the forensic lab participating in the project.

For each genuine individual, the tool generates a SVM trained to recognize his/her Connected Components. In the identification process the questioned individual specimens are presented to every genuine SVM. A similarity rate or score is computed for each genuine user and the final classification is defined as the highest genuine user's. Table 1 resumes the results.

In this preliminary experiment, this writer identification system shows a 80% identification accuracy with 20 users in the database (10 genuine individuals and 10 questioned individuals). Two errors were observed for questioned users "q07" and "q08" during the experiment. These mistakes appear to be caused by the SVM of the genuine user "svm05", whose training process was not as efficient as in the other nine cases. A high scoring rate is produced by this SVM for all the questioned users, and this issue has a particular effect for cases "q07" and "q08" where the "svm05" wins versus the other SVM modules and is wrongly selected as the winner.

Table 1. Identification results.

%	q01	q02	q03	q04	q05	q06	q07	q08	q09	q10
Svm01	71	25	25	24	33	24	28	26	31	24
Svm02	5	47	28	20	21	9	32	18	16	29
Svm03	47	17	37	33	28	49	25	43	44	25
Svm04	28	28	19	68	32	10	21	45	6	22
Svm05	32	32	28	55	44	14	38	46	13	32
Svm06	4	28	10	3	8	58	13	4	51	4
Svm07	7	4	21	15	16	4	33	12	3	20
Svm08	42	12	20	11	25	44	16	35	30	11
Svm09	8	17	25	13	19	49	18	18	58	11
Svm10	12	34	31	19	19	10	29	20	16	64

Ok	q01-10	%score obtained with the questioned users 1-10
Error	svm01-10	genuine SVMs

4 Conclusion

The analysis of these results with a limited database of individuals suggests that the basic idea of this research can be effectively exploded in order to develop automatic writer identification tools based on Connected Components analysis. Moreover, multiple-character connected components are quite useful in writer identification. As it was explained above, the "syllabic" impulse is an important feature in a specific writer style. Therefore, the use of multiple-character connected components may increase recognition performance much in the same way as an expert might. Next experiments will be directed to extend the number of individuals involved in them, in order to get a larger database. With this extended database the results of the experiments will achieve high accuracy and will provide more information about the Connected Components distribution on large populations.

Future work on this system will improve the data collection method in this Connected Components analysis approach. Now the tool generates thousands of writing specimens (i.e. Connected Components) for a reduced group of users, but in order to use this kind of application in large-scale situations, the amount of data must be reduced to achieve reasonable response times. Clustering and data mining techniques will enable the application to use only relevant specimens and discard noise samples.

Acknowledgment. The authors would like to thank all the Guardia Civil Corps for their valuable help. Specially, we would like to mention the staff at 'Laboratorio de Grafística', from 'Servicio de Policía Judicial' at 'Dirección General de la Guardia Civil'. This work was supported in part by the Spanish Government (MCYT) under Project FIT-070200-2002-131.

References

1. H. Hardy and W. Fagel, "Methodological aspects of handwriting identification," *Journal of Forensic Document Examination*, Fall, 1995.
2. R. A. Huber and A. M. Headrick, *Handwriting identification: facts and fundamentals*. Ed. CRC Press, 1999.
3. A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics. Personal identification in networked society*. Ed. Kluwer Academic Publishers, 1999.
4. D. D. Zhang, *Automated biometrics. Technologies and systems*. Ed. Kluwer Academic Publishers, 2000.
5. S. N. Srihari *et al.*, "Handwriting identification: research to study validity of individuality of handwriting and develop computer-assisted procedures for comparing handwriting," Tech. Rep. CEDAR-TR-01-1, Center of Excellence for Document Analysis and Recognition, University at Buffalo, State University of New York, Feb. 2001, 54 pp.
6. H. E. S. Said, G. S. Peake, T. N. Tan and K. D. Baker, "Writer identification from non-uniformly skewed handwriting images," in *Proc. 9th. British Machine Vision Conference*, 2000, pp. 478-487.
7. W. Kuckuck, "Writer recognition by spectra analysis," in *Proc. Int. Conf. In Security Through Science Engineering,* 1980, West Berlin, Germany, pp. 1-3.

8. S. N. Srihari, S. H. Cha, H. Arora and S. Lee, “Individuality of handwriting,” in *Proc. 6th. Int. Conf. on Document Analysis and Recognition (ICDAR’01),* Seattle, WA, Sep. 2001, pp. 106-109.
9. S. H. Cha and S. N. Srihari, “Multiple feature integration for writer verification,” in *Proc. 7th. Int. Workshop on Frontiers of Handwriting Recognition (IWFHR2000),* Amsterdam, The Netherlands, Sep. 2000, pp. 333-342.
10. B. Zhang, S. N. Srihari, S. Lee, “Individuality of handwritten characters,” in *Proc. 7th. Int. Conf. on Document Analysis and Recognition (ICDAR’03),* Edinburgh, Scotland, Aug. 2003, paper id 527.
11. S. Impedovo, *Fundamentals in Handwriting Recognition*, Ed. Berlin Heidelberg: Springer-Verlag, 1994.
12. R. C. Gonzalez, R. E. Woods, *Digital Image Processing*. Ed. Addison-Wesley, 1992.
13. N. Otsu, “A threshold selection method from gray-scale histogram,” *IEEE Transactions System, Man and Cybernetics,* vol. 9, pp. 62-66, Jan. 1979.
14. R. N. Morris, “Forensic Handwriting Identification:fundamental concepts and principles”. Ed. Academic Press, 2000.
15. V. Vapnik. “The Nature of Statistical Learning Theory”. Springer-Verlag, New York, 1995.
16. C. Burges and B. Schölkopf. “*Improving the accuracy and speed of support vector learning machines*”. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pp 375-381, Cambridge, MA, 1997. MIT Press.
17. C. Burges. “*A Tutorial on Support Vector Machines for Pattern Recognition*”. *Knowledge Discovery and Data Mining*, 2(2), pp 121-167, 1998.

Modeling Intrusion Detection Systems Using Linear Genetic Programming Approach

Srinivas Mukkamala[1], Andrew H. Sung[1], and Ajith Abraham[2]

[1] Department of Computer Science, New Mexico Tech, Socorro, NM 87801
{srinivas,sung}@cs.nmt.edu
[2] Department of Computer Science, Oklahoma State University, Tulsa, OK 74106
ajith.abraham@ieee.org

Abstract-This paper investigates the suitability of linear genetic programming (LGP) technique to model efficient intrusion detection systems, while comparing its performance with artificial neural networks and support vector machines. Due to increasing incidents of cyber attacks and, building effective intrusion detection systems (IDSs) are essential for protecting information systems security, and yet it remains an elusive goal and a great challenge. We also investigate key feature indentification for building efficient and effective IDSs. Through a variety of comparative experiments, it is found that, with appropriately chosen population size, program size, crossover rate and mutation rate, linear genetic programs could outperform support vector machines and neural networks in terms of detection accuracy. Using key features gives notable performance in terms of detection accuracies. However the difference in accuracy tends to be small in a few cases.

1 Introduction

Since most of the intrusions can be located by examining patterns of user activities and audit records, many IDSs have been built by utilizing the recognized attack and misuse patterns. IDSs are classified, based on their functionality, as misuse detectors and anomaly detectors. Misuse detection systems use well-known attack patterns as the basis for detection [1,2]. Anomaly detection systems make use user profiles as the basis for detection; any deviation from the normal user behavior is considered an intrusion. One of the main problems with IDSs is the overhead, which can become unacceptably high. To analyze system logs, the operating system must keep information regarding all the actions performed, which invariably results in huge amounts of data, requiring disk space and CPU resource. Next, the logs must be processed to convert into a manageable format and then compared with the set of recognized misuse and attack patterns to identify possible security violations. Further, the stored patterns need be continually updated, which would normally involve human expertise. An intelligent, adaptable and cost-effective tool that is capable of (mostly) real-time intrusion detection is the goal of the researchers in IDSs. Various AI techniques have been utilized to automate the intrusion detection process to reduce human intervention; several such techniques include neural networks [3,4,5], and machine learning [5,6]. Several data mining techniques have been introduced to identify key features or parameters that define intrusions [6,7].

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 633-642, 2004.

LGP has been already successfully implemented to a variety of machine learning problems [8]. This paper investigates the suitability of linear genetic programming technique, for modeling intrusion detection systems and indentifying the key features that help in deciding whether a connection is intrusive or normal activity. We also compare the performance of LGP with Support Vector Machines (SVM) and a Neural Network (NN) trained using resilient backpropagation learning. Performance metrics include a few critical aspects of intrusion detection like training and testing times, scalability and detection accuracy that help IDSs perform in real time or near real time. The data we used in our experiments originated from MIT's Lincoln Lab [9].

We perform experiments to classify the network traffic patterns according to a 5-class taxonomy. The five classes of patterns in the DARPA data are (normal, probe, denial of service, user to super-user, and remote to local). The experimental results of overall classification accuracy and class specific accuracies using linear genetic programs, support vector machines and resilient back propagation neural network are reported. Results obtained form feature ranking experiments are also reported with brief description of the most important features for each class.

In Section 2 a brief introduction to linear genetic programs, support vector machines and resilient back propagation algorithm is given. Section 3 briefly introduces the data we used to compare performance of different soft computing techniques. Section 4 presents the experimental results using LGP, SVM and NN. We briefly describe the feature selection/importance of ranking and the related results in section 5. The summary and conclusions of our work are given in section 6.

2 Linear Genetic Programming (LGP)

Linear genetic programming is a variant of the GP technique that acts on linear genomes [10,11]. Its main characteristics in comparison to tree-based GP lies in that the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like C/C ++). An alternate approach is to evolve a computer program at the machine code level, using lower level representations for the individuals. This can tremendously hasten up the evolution process as, no matter how an individual is initially represented, finally it always has to be represented as a piece of machine code, as fitness evaluation requires physical execution of the individuals. The basic unit of evolution is a native machine code instruction that runs on the floating-point processor unit (FPU). Since different instructions may have different sizes, here instructions are clubbed up together to form instruction blocks of 32 bits each. The instruction blocks hold one or more native machine code instructions, depending on the sizes of the instructions. A crossover point can occur only between instructions and is prohibited from occurring within an instruction. However the mutation operation does not have any such restriction.

2.1 Resilient Backpropagation (RBP)

The purpose of the resilient backpropagation training algorithm is to eliminate the harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor whenever the derivative with respect that weight changes sign from the previous iteration. If the derivative is zero, then the update value remains the same. Whenever the weights are oscillating the weight change will be reduced. If the weight continues to change in the same direction for several iterations, then the magnitude of the weight change will be increased [12].

2.2 Support Vector Machines (SVMs)

The SVM approach transforms data into a feature space *F* that usually has a huge dimension. It is interesting to note that SVM generalization depends on the geometrical characteristics of the training data, not on the dimensions of the input space [13,14]. Training a support vector machine (SVM) leads to a quadratic optimization problem with bound constraints and one linear equality constraint. Vapnik shows how training a SVM for the pattern recognition problem leads to the following quadratic optimization problem [15].

Minimize: $$W(\alpha) = -\sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \quad (1)$$

Subject to $$\sum_{i=1}^{l} y_i \alpha_i \quad (2)$$

$$\forall i : 0 \le \alpha_i \le C$$

Where l is the number of training examples α is a vector of l variables and each component α_i corresponds to a training example (x_i, y_i). The solution of (1) is the vector α^* for which (1) is minimized and (2) is fulfilled.

3 Intrusion Detection Data

In the 1998 DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks [16,17]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [6]. Of this database a subset of 494021 data were used, of which 20% represent normal patterns.

Attack types fall into four main categories namely (1) Probing: surveillance and other probing (2) DoS: denial of service (3) U2Su: unauthorized access to local super user (root) privileges and (4) R2L: unauthorized access from a remote machine

3.1 Probing

Probing is a class of attacks where an attacker scans a network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits. There are different types of probes: some of them abuse the computer's legitimate features; some of them use social engineering techniques. This class of attacks is the most commonly heard and requires very little technical expertise.

3.2 Denial of Service Attacks

Denial of Service (DoS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine. There are different ways to launch DoS attacks: by abusing the computers legitimate features; by targeting the implementations bugs; or by exploiting the system's misconfigurations. DoS attacks are classified based on the services that an attacker renders unavailable to legitimate users.

3.3 User to Root Attacks

User to root (U2Su) exploits are a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and environment assumptions.

3.4 Remote to User Attacks

A remote to user (R2L) attack is a class of attacks where an attacker sends packets to a machine over a network, then exploits machine's vulnerability to illegally gain local access as a user. There are different types of R2U attacks; the most common attack in this class is done using social engineering.

4 Experiments

In our experiments, we perform 5-class classification. The (training and testing) data set contains 11982 randomly generated points from the data set representing the five classes, with the number of data from each class proportional to its size, except that the smallest class is completely included. The set of 5092 training data and 6890 testing data are divided in to five classes: normal, probe, denial of service attacks, user to super user and remote to local attacks. Where the attack is a collection of 22 different types of instances that belong to the four classes described in section 3, and the other is the normal, and the other is the normal data. The normal data belongs to class1, probe belongs to class 2, denial of service belongs to class 3, user to super user belongs to class 4, remote to local belongs to class 5. Note two randomly generated separate data sets of sizes 5092 and 6890 are used for training and testing the LGPs, SVMs, and RBP respectively. Same training and test datasets were used for all the experiments. Tables 1,2 and 3 summarize the overall classification accuracy of LGPs, RBP and SVMs.

4.1 Linear Genetic Programming Training

LGP manipulates and evolves program at the machine code level [10]. The settings of various LGP parameters are of utmost importance for successful performance of the system. This section discusses the different parameter settings used for the experiment, justification of the choices and the significances of these parameters. The population space has been subdivided into multiple subpopulation or demes. Migration of individuals among the subpopulations causes evolution of the entire population. It helps to maintain diversity in the population, as migration is restricted among the demes. Moreover, the tendency towards a bad local minimum in one deme can be countered by other demes with better search directions. The various LGP search parameters are the mutation frequency, crossover frequency and the reproduction frequency: The crossover operator acts by exchanging sequences of instructions between two tournament winners. A constant crossover rate of 90% has been used for all the simulations. After a trial and error approach, the parameter settings in Table 1 are used to develop IDS. Five LGP models are employed to perform five class classifications (normal, probe, denial of service, user to root and remote to local). We partition the data into the two classes of "Normal" and "Rest" (Probe, DoS, U2Su, R2L) patterns, where the Rest is the collection of four classes of attack instances in the data set. The objective is to separate normal and attack patterns. We repeat this process for all classes. Table 1 summarizes the results of the experiments using LGPs.

Table 1. Performance of LGPs

Class	Population Size	Program Size	Crossover Rate	Mutation Rate	Testing Accuracy (%)
Normal	1024	256	72.24	90.51	99.89
Probe	2048	512	51.96	83.30	99.85
DoS	2048	512	71.15	78.14	99.91
U2Su	2048	256	72.24	90.51	99.80
R2L	2048	512	71.15	78.14	99.84

4.2 RBP Experiments

In our study we used two hidden layers with 20 and 30 neurons each and the network is trained using resilient backpropogation neural network. As multi-layer feed forward networks are capable of multi-class classifications, we partition the data into 5 classes (Normal, Probe, Denial of Service, and User to Root and Remote to Local).We used testing data set (6890), network architecture [41 20 30 1] to measure the peformace of the RBP. The top-left entry of Table 2 shows that 1394 of the actual "normal" test set were detected to be normal; the last column indicates that 99.6 % of the actual "normal" data points were detected correctly. In the same way, for "Probe" 649 of the actual "attack" test set were correctly detected; the last column indicates that 92.7% of the actual "Probe" data points were detected correctly. The bottom row shows that 96.4% of the test set said to be "normal" indeed were "normal" and 85.7% of the test set classified, as "probe" indeed belongs to Probe. The overall accuracy of the classification is 97.04 with a false positive rate of 2.76% and false negative rate of 0.20.

Table 2. Performance of resilient back propagation neural network

	Normal	Probe	DoS	U2Su	R2L	%
Normal	1394	5	1	0	0	99.6
Probe	49	649	2	0	0	92.7
DoS	3	101	4096	2	0	97.5
U2Su	0	1	8	12	4	48.0
R2L	0	1	6	21	535	95.0
%	96.4	85.7	99.6	34.3	99.3	

4.3 SVM Experiments

Because SVMs are only capable of binary classifications, we employed five SVMs, for the 5-class classification problem.. We partition the data into the two classes of "Normal" and "Rest" (Probe, DoS, U2Su, R2L) patterns, where the rest is the collection of four classes of attack instances in the data set. The objective is to separate normal and attack patterns. We repeat this process for all classes. Training is done using the RBF (radial bias function) kernel option; an important point of the kernel function is that it defines the feature space in which the training set examples will be classified. Table 3 summarizes the results of the experiments using SVMs.

Table 3. Performance of SVMs

Class	Training time (sec)	Testing time (sec)	Accuracy (%)
Normal	7.66	1.26	99.55
Probe	49.13	2.10	99.70
DoS	22.87	1.92	99.25
U2Su	3.38	1.05	99.87
R2L	11.54	1.02	99.78

5 Ranking the Significance of Features

Feature selection and ranking [16,17] is an important issue in intrusion detection. Of the large number of features that can be monitored for intrusion detection purpose, which are truly useful, which are less significant, and which may be useless? The question is relevant because the elimination of useless features (the so-called audit trail reduction) enhances the accuracy of detection while speeding up the computation, thus improving the overall performance of an IDS. The feature ranking and selection problem for intrusion detection is similar in nature to various engineering problems that are characterized by:

a. Having a large number of input variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of varying degrees of importance to the output $\mathbf{y}$; i.e., some elements of $\mathbf{x}$ are essential, some are less important, some of them may not be mutually independent, and some may be useless or irrelevant (in determining the value of **y**)

b. Lacking an analytical model that provides the basis for a mathematical formula that precisely describes the input-output relationship, $\mathbf{y} = \boldsymbol{F}(\mathbf{x})$

c. Having available a finite set of experimental data, based on which a model (e.g. neural networks) can be built for simulation and prediction purposes

Due to the lack of an analytical model, one can only seek to determine the relative importance of the input variables through empirical methods. A complete analysis would require examination of all possibilities, e.g., taking two variables at a time to analyze their dependence or correlation, then taking three at a time, etc. This, however, is both infeasible (requiring 2n experiments!) and not infallible (since the available data may be of poor quality in sampling the whole input space). Features are ranked based on their influence towards the final classification. Once the rank of the features is obtained unimportant features in groups of five are deleted and the performace is evaluated. Figures 1 and 2 gives the rank and importance towards calssification of each class respectively. Table 4 gives a brief description of the most important five features for each class. Feature ranking performance results for the individual classes are given in figures 3,4,5,6, and 7.

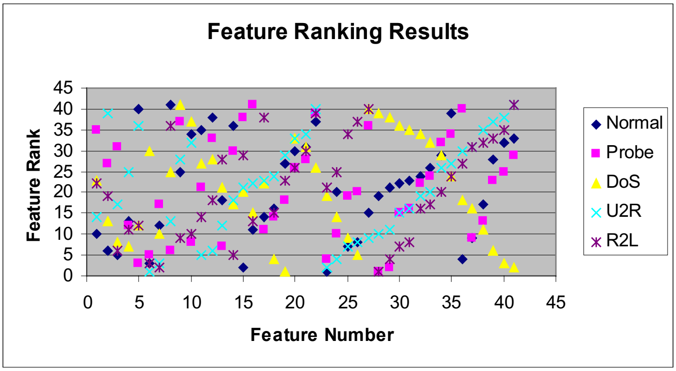


Fig. 1. Rank of each features for each class

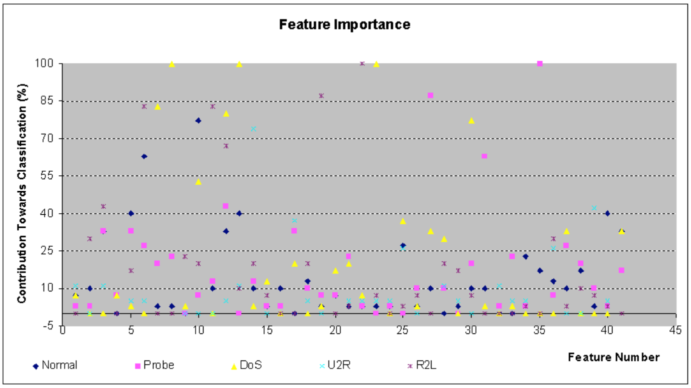


Fig. 2. Feature importance towards classification for each class

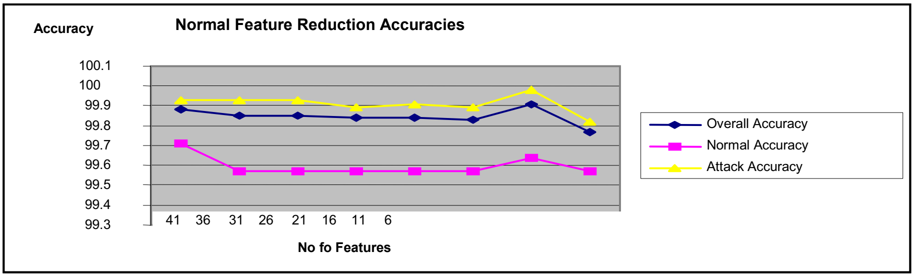


Fig. 3. Feature reduction results for normal

Table 4. Description of 5 most important features

Class	Feature Discription
Normal	▪ hot indicators: Number of "hot" indicators ▪ destination bytes: Number of bytes sent from the destination system to the host system ▪ source bytes: Number of bytes sent from the host system to the destination system ▪ compromised conditions: Number of compromised conditions ▪ dst_host_rerror_rate: % of connections that have REJ errors from a destination host
Probe	▪ dst_host_diff_srv_rate: % of connections to different services from a destination host ▪ rerror_rate: % of connections that have REJ errors ▪ srv_diff_host_rate: % of connections that have same service to different hosts ▪ logged in: binary decision ▪ service: type of service
DoS	▪ count: Number of connections made to the same host system in a given interval of time ▪ compromised conditions: Number of compromised conditions ▪ wrong_fragments: no of wrong fragments ▪ land: 1 if connection is from/to the same host/port; 0 otherwise ▪ logged in: 1 if successfully logged in; 0 otherwise
U2Su	▪ root shell: 1 if root shell is obtained; 0 otherwise ▪ dst_host_srv_serror_rate: % of connections to the same service that have SYN errors from a destination host ▪ no of file creations: no of file creation operations ▪ serror_rate: % of connections that have SYN errors ▪ dst_host_same_src_port_rate: % of connections to same service ports from a destination host
R2L	▪ guest login: 1 if the login is a "guest' login; 0 otherwise ▪ no of file access: no of operations on access control files ▪ destination bytes: Number of bytes sent from the destination system to the host system ▪ failed logins: no of failed login attempts ▪ logged in: binary decision

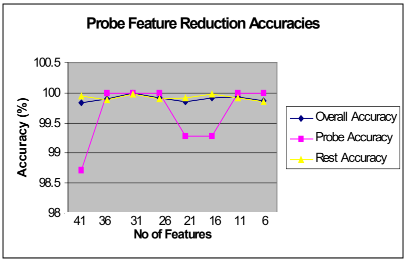


Fig. 4. Feature reduction results for probe

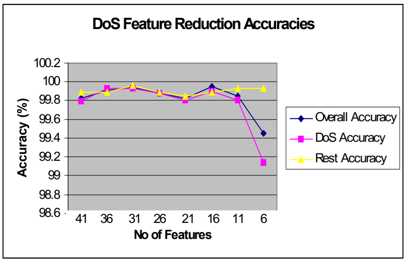


Fig. 5. Feature reduction results for DoS

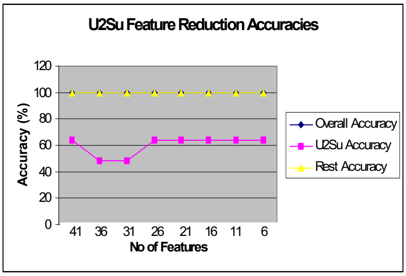


Fig. 6. Feature reduction results for U2Su

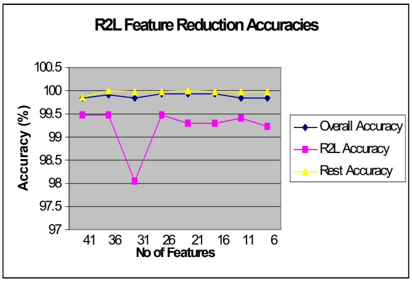


Fig. 7. Feature reduction results for R2L

6 Conclusions

Table 5 summarizes the overall performance of the three soft computing paradigms considered. LGPs outperform SVMs and RBP in terms of detection accuracies with the expense of time. SVMs outperform RBP in the important respects of scalability (SVMs can train with a larger number of patterns, while would ANNs take a long time to train or fail to converge); training time and prediction accuracy. Resilient back propagation achieved the best performance among the several other neural network learning algorithms we considered in terms of accuracy (97.04 %). The performances of using the important features for each class, give comparable performance with no significant differences, to that of using all 41 features.

Table 5. Performance comparison of testing for class specific classification

Class	SVMs Accuracy (%)	RBP Accuracy (%)	LGP Accuracy (%)
Normal	98.42	99.57	99.64
Probe	98.57	92.71	99.86
DoS	99.11	97.47	99.90
U2Su	64	48	64
R2L	97.33	95.02	99.47

We note, however, that the difference in accuracy figures tend to be very small and may not be statistically significant, especially in view of the fact that the 5 classes of patterns differ in their sizes tremendously. More definitive conclusions can only be made after analyzing more comprehensive sets of network traffic data.

References

1. Denning D. (1987) “An Intrusion-Detection Model,” *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 2, pp.222-232.
2. Kumar S., Spafford E. H. (1994) “An Application of Pattern Matching in Intrusion Detection,” Technical Report CSD-TR-94-013. Purdue University.
3. Cannady J. (1998) “Applying Neural Networks for Misuse Detection,” *Proceedings of 21st National Information Systems Security Conference*, pp.368-381.
4. Ryan J., Lin M-J., Miikkulainen R. (1998) “Intrusion Detection with Neural Networks,” *Advances in Neural Information Processing Systems,* Vol. 10, Cambridge, MA: MIT Press.
5. Mukkamala S., Janoski G., Sung A. H. (2002) “Intrusion Detection Using Neural Networks and Support Vector Machines,” *Proceedings of IEEE International Joint Conference on Neural Networks*, pp.1702-1707.
6. Stolfo J., Wei F., Lee W., Prodromidis A., and Chan P. K. (1999) “Cost-based Modeling and Evaluation for Data Mining with Application to Fraud and Intrusion Detection,” *Results from the JAM Project by Salvatore.*
7. Mukkamala S., Sung A. H. (2002) “Identifying Key Features for Intrusion Detection Using Neural Networks,” *Proceedings of ICCC International Conference on Computer Communications*, pp. 1132-1138.
8. Banzhaf. W., Nordin. P., Keller. E. R., Francone F. D. (1998) “Genetic Programming : An Introduction on The Automatic Evolution of Computer Programs and its Applications,” Morgan Kaufmann Publishers, Inc.
9. http://www.ll.mit.edu/IST/ideval/data/data_index.html
10. AIMLearning Technology, http://www.aimlearning.com.
11. Brameier. M., Banzhaf. W. (2001) “A comparison of linear genetic programming and neural networks in medical data mining, Evolutionary Computation,” IEEE Transactions on, Volume: 5(1), pp. 17-26.
12. Riedmiller M., and Braun H. (1993) “A direct adaptive method for faster back propagation learning: The RPROP algorithm”, *Proceedings of the IEEE International Conference on Neural Networks.*
13. Joachims T. (1998) “Making Large-Scale SVM Learning Practical,” LS8-Report, University of Dortmund, LS VIII-Report.
14. Joachims T. (2000) “SVMlight is an Implementation of Support Vector Machines (SVMs) in C,” http://ais.gmd.de/~thorsten/svm_light. University of Dortmund. Collaborative *Research Center on Complexity Reduction in Multivariate Data (SFB475*).
15. Vladimir V. N. (1995) “The Nature of Statistical Learning Theory,” *Springer.*
16. Kendall K. (1998) “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems”, *Master's Thesis, Massachusetts Institute of Technology.*
17. Webster S. E. (1998) “The Development and Analysis of Intrusion Detection Algorithms,” *M.S. Thesis, Massachusetts Institute of Technology.*

Data Mining in Evaluation of Internet Path Performance

Leszek Borzemski

Wroclaw University of Technology,
Institute of Control and Systems Engineering,
Wybrzeze Wyspianskiego 27,
50-370 Wroclaw, Poland
leszek@ists.pwr.wroc.pl

Abstract. The data mining algorithms and tools have been employed in many engineering and non-engineering applications. This paper proposes a new practical application of data mining in the study of problems related to the Internet performance. Well-predicted performance is a key engineering issue in Internet-based processing, e.g. in peer-to-peer systems. The measurements of round-trip-time performed by the traceroute probing technique are used for Internet path performance evaluation. We show how data mining can be used by the end-users to evaluate the performance of the communication path between their location and a specific Internet host in a long-term performance prediction scale. The mining data is mined using the IBM Intelligent Miner for Data system. We discover how the round-trip times of the packets and the number of hops they pass vary with the day of the week and the time of the measurement and use this knowledge to built the decision tree that can be a useful guide to the future characteristics of relevant properties of given Internet path in a long-term scale.

1 Introduction and Related Work

The usage of Internet services grows so rapidly that both service providers and users are urgently seeking new methods and tools in order to predict end-to-end performance. Clients perceive good Internet performance as low latency, high throughput and high availability. Internet quality of service is extremely difficult to study in an integrated way. It has never been easy to determine whether slow responses are due to either network problems or end-system problems on both sides, and both. Most of these performance problems are transient and very complex in the relationships between different factors that may influence each other. Therefore we cannot exactly diagnose and isolate their key sources. Since almost 60% latency, as perceived by end-users at their microscopic level, refers to the end-to-end path latency [5], therefore understanding the network performance problems is very important. New network applications that are becoming main contributors in Internet traffic, such as peer-to-peer and grid applications [16], require the predictable performance behavior of Internet paths used for communication. They may need both short-term and long-term performance forecasts. Short-term forecasting requires instantaneous measuring of network resource performance and usage. In long-term forecasting we may stop thinking about

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 643-652, 2004.

instantaneous measuring and may try to predict path performance characteristics from the measurement data describing path performance which was collected for some time period in the past.

Characterizing the behavior patterns of the end-to-end Internet paths is critical for diagnosing performance problems. Unfortunately, the Internet was not primarily designed with the goal to help the end-users in the measurements and evaluation of their Internet paths. For short-time evaluation the user usually use the ping utility program to diagnose the network latency. Though the user can do it before transmitting, this is the knowledge about only current value of latency and none knowledge is about the future characteristics of the latency. Transport network protocols like TCP keep an estimate of the current RTT on each connection, however the user application cannot get the RTT value from TCP because and it is internal to TCP. In such case, the user can use one of the special forecasting services such as well-known Network Weather Service (NWS) [19]. NWS is a distributed framework that aims to provide short-term forecasts of dynamically changing performance characteristics of a distributed set of Internet hosts. It measures, among other parameters, the end-to-end network latency and network bandwidth. It periodically monitors and dynamically forecasts short-term expected performance of the network path between hosts. In order to calculate expected performance it runs all the time and instantaneously gathers readings of performance conditions from a set of monitors. Unfortunately, NWS runs only in Unix operating system environments and does not support long-term forecasts.

Many complex measurement activities and projects have been launched on the Internet [1, 2, 3, 4, 8, 13, 14, 19, 20, 21]. They present the network weather reports but most of them are aimed at dealing with the performance problem related to the whole Internet or a significant part of it, where large amounts of measured data regarding, for instance, round trip delay among several node pairs over a few hours, days or months, are collected. For these purposes the specific measurements and data analysis infrastructures are used. Such measurements are mainly performed for large Internet-based projects by means of complex measuring infrastructure and especially they are performed in the core of Internet. But how can we characterize end-user connectivity not included in any of the aforementioned projects? The answer would be to track our way.

Measurements can merely report the network state at the time of the measurement. They are effectively used by various protocols to test and monitor current performance and to take necessary action when changes are detected. Nevertheless, when the relevant network properties exhibit to be constant (stable) over observed network life-time span, then the measurements can be also a useful guide to the future. The concept of constant (steady) network parameters is especially more useful for coarser time scale than for fine time scales. It was shown in [20] that in general the round trip packet delay as measured by the Round-Trip-Time (RTT) appears well described as steady on time scale of 10-30 minutes. In our paper we also use RTT as performance characteristics of Internet path.

Internet periodic behavior is observed. It is evidenced by web and network event logs. Paper [12] shows that the number of HTTP operations per second is nonstationary and its five-minute mean changes essentially with time-of-day and day-of-week. It was also shown [12] that time-of-day explains 53% of the variability in the raw data of HTTP transactions collected over eight months from a production Web server whereas after including day-of-week and month factors we reach only 64% of the variability in the data. Our user is interested only in a general path characteristic. Therefore, we simplify matters in our further analysis by only considering time-of-day and day-of-week factors. Also we are conscious that many network events may effect on path performance. Most of them are beyond our control and knowledge, however some knowledge of them (such as regular network measurements obtained from the Internet weather service [4, 6, 7] may be included in a knowledge-based data mining. This knowledge is not considered here. We use only the knowledge which can be derived from our measurement data.

We are not concerned in uncovering the basic nature of the underlying phenomena. Instead the focus is on producing a solution that can generate useful predictions. We do not want to predict the particular value of path performance metrics (e.g. RTT value) at specific time frame but we can try to classify the "network weather" that may occur in a long-time scale. Therefore, the data mining is proposed as an information tool that can guide decisions in conditions of limited certainty.

We assume that the end-user is a domain expert and is not either specialist in statistics or Internet but he/she knows crucial concepts in data mining and can use a software tools for data mining. We want ultimately to create a prediction model fully interpretable by domain expert who based on our method will be able to schedule his/her Internet activity (e.g. data transfers) using the particular network path. We propose such a method which is easy enough, yet powerful enough and usable by the user.

In this paper we assume that the path performance behavior can be classified into some typical categories (classes, patterns). Can we develop a general characterization of different performance behavior types? Can we develop a model which will classify new data and predict class labels? We do not know if there are any different performance pattern types at all. If there are different performance pattern types, we do not know how many types there are. In our problem patterns and relationships in data are not obvious. This paper proposes to address this problem with a data mining that combines clustering and classification functions.

We want to find the differences between these classes especially in terms of the moments of measurements (based on time-of-day and day-of-week factors) and to create a model describing the segmentation of the data. The model should explain the rules grouping measurement data records with similar properties and could be used to decide what class will occur for the specified measurement moment in the future. In this sense we can think about a long-term forecast of future Internet path behavior. Such knowledge may be used for scheduling of future usage of Internet path for a particular time and date. The user may want to schedule Internet data file transfers in a long-term scale but, unfortunately, cannot reserve a bandwidth of network connection (i.e. QoS) and therefore wants to know Internet path behavior in the demanded time period using previous historical measurements.

Interest in data mining is motivated by the growth of computerized data sets and by the high potential value of patterns discovered in those data collections. As for today Internet data mining is mainly devoted to Web mining [17]. We can find several data mining applications in science, engineering, business, industry and medicine [11].

Data mining and knowledge discovery in computer networks are of great topical interest of early works [10, 13]. Nowadays the classical statistical data analysis is usually performed to derive network performance characteristics [2, 3, 4, 6, 15]. This contribution shows that data mining methods can be used for that analysis for long-term network behavior. Our strategy involves discovering knowledge that may characterize performance behavior of Internet paths and then making use of this knowledge to guide future user actions.

This paper describes an ongoing Internet performance forecasting project. Path performance is characterized by the end-to-end latency and measured by the Round-Trip-Time. The traceroute packet probing technique is used for measurements. The ultimate goal is to use a commercial data mining system providing the environment layer API functions for controlling the execution of mining runs and results. The environment layer API gives a possibility to develop a system for (semi)-automated analysis. Therefore we used the IBM Intelligent Miner for Data and its clustering and classification mining functions.

The remainder of this paper is organized as follows. Section 2 presents our measurement and analysis methodology. The results of mining using sample data are discussed in Sections 3 and 4. Finally, concluding remarks appear in Section 5.

2 Measurement and Analysis Methodology

Our aim was to mine a two-way connectivity between pairs of hosts in the Internet. In every pair the source host was installed in our local network whereas its counter partners called target hosts were somewhere in the Internet. The data mining sets used here were compiled from the measurements performed during six months in 2003, starting from 26 February 2003. The measurements were performed periodically each half an hour and probes were sent between our source host and 18 servers taken from SLAC list [21]. Hence we collected information from 18 Internet paths, each in both directions.

We used the traceroute as a tool for Internet network connectivity and routing testing from our source server [13]. It is a standard tool available on most Internet hosts and is flexible and informative for common user. We have chosen this tool due its simplicity and low network traffic overhead. A number of research groups used also this tool, for example for generating maps of Internet [15]. Traceroute is the program that tests the route over the network between two destinations on the Internet and lists all the intermediate routers (nodes) a testing packet must pass through to rich its destination as well as the latency between the source host and intermediate systems. This latency called the Round-Trip Time (RTT) tells us how long a packet goes from a source host to relevant remote system and back again. RTT is a common measure of the current delay on a network. RTT can be measured by the ping tool but we used the traceroute to measure RTT and to discover IP routing. Traceroute operates only in one direction, i.e. from traceroute source host to traceroute target host. Any connection over the Internet actually depends on two routes: the route from source host to the target host, and the route from target host back to source host. These routes may be (and often are) different (asymmetric). To discover reverse routing, we used reverse traceroute that is simply the traceroute but diagnosing the network in an opposite direction, i.e. from the target host to the source host. To proceed with our task we have chosen such service of reverse traceroute servers offered by these 18 target servers in the SLAC project [21]. Using traceroute to remote host and reverse traceroute from remote host to our host we measured two-way latency defined by both RTTs shown by traceroute and reverse traceroute.

In this paper we present the results of measurements and performance evaluation made using data mining methods for the path between the source host operating within the network of the Wroclaw University of Technology and a target host randomly selected from available target servers. As the data mining system we used the IBM Intelligent Miner for Data 8.1 [18] running on the IBM RISC/6000 F80 computer along with AIX 4.3.3 and DB2 8.1 database. The measurements were collected by this server. Data mining was also done on PC-based host equipped with P4 2.4 GHz, 512 MB RAM, 120 GB HDD, MS Windows XP Professional PL and DB2 8.1 database. The measurements are collected in the relational table where the rows contained all the data collected for the relevant path measured at the specific time. Each record has 21 fields – among them there are the following fields: SOURCE_HOST_NAME, HOST_NAME, IP_ADDRESS, MINUTE, HOUR, MONTH, YEAR, DAY, TIME_STAMP, RTT_1, RTT_2, RTT_3, RTT, and HOP. RTT is the average RTT calculated from RTT_1, RTT_2 and RTT_3. The HOST_NAME is the name of either some intermediate router tracerouted on the way to the target server, or the target server itself.

We propose to use a clustering data mining function followed by classification function data mining in such a way that the results of clustering are the inputs to classification. Clustering approaches address segmentation problems and assign data records with a large number of attributes into a set of groups of records called "clusters". This process is performed automatically by clustering algorithms that identify the distinguishing characteristics of the dataset and then partition the n-dimensional space defined by the dataset attributes along natural cleaving boundaries. There is no need to identify the groupings desired or the attributes that should be used to segment the dataset. Clustering segments data records into groups, i.e. clusters that are classes of network behavior. Each cluster labeled by its ID describes some discovered characteristics of path performance.

Because cluster definitions are not easily extracted and easily interpreted by the user, we prepare cluster characterization using a decision tree. The results of clustering are the inputs to classification. Building a decision tree with the cluster ID as the field to be classified and using training data allows explicit rules to be extracted for each cluster. Therefore, in the next stage we construct a decision tree model to be used in classifying new data. Classification function is used to develop a predictive model. Model is developed in two phases: training and testing. In training phase we develop a model using some historical data, whereas in testing phase we try out the model on new input data.

There are the following steps in the data mining process: (i) Data selection, (ii) Data preparation and transformation, (iii) Clustering, (iv) Cluster result analysis, (v) Cluster characterization using a decision tree. In the data selection step we defined a relational table discussed earlier. We have measured and collected data over six months of periodic active experiments for all target hosts in both directions using the traceroute facility. The performance of each end-to-end network path (the direction of the path is important) is analyzed individually; therefore further analysis is applied only to a single (directed) path from one host to another. In this paper we choose the path from our local host to a single target host. Therefore, for further analysis only data records matching that path were selected from the table. In the second step we had to clean the data. There were missing RTT values – we assigned these missing values a mean value calculated on the basis of previous and next observations. For every data record we calculated an average RTT based on three probes (we do not consider invalid RTT values greater than 10000 ms, that shows host/network failure). RTT is a numeric continuous variable therefore before applying the clustering algorithm we made the discretization of RTT data using equal-width (except for the last bin) discretization (binning) scheme. The following breakpoints were used in the discretization of RTT data: 0, 25, 50, 75, 100, 125, 150,…, 325, 350, 10000, so data was divided into 15 ordinal values. We also mapped values of RTT to low, medium, or high, as they will be used in interpretation of results. For example: value RTT is medium, or between 50 and 75. Other fields used in mining were: the day of the week DAY € {1, 2, 3, 4, 5, 6, 7}, the hour of the measurement HOUR € {0, 1, 2, 3… 23} and the length of the path between source and target host – the number of hops HOP € {1, 2…}.

3 Application of Clustering Mining Function

We want to find some differences between these groups especially in terms of the time of measurements and to create a model describing the factors responsible for such a split of records called clusters. The model should explain the rules of grouping records with similar properties and based on the assigned group description it would give the information about the results most probable to achieve at a given moment in the future without performing the measurements. For finding the groups of similar measurement results in our input mining base we used the neural clustering algorithm, which employs a Kohonen Feature Map neural network [18]. This method uses a process of self-organization to group similar input records. The user specifies the number of clusters and the maximum number of passes trough the data. Specifying multiple passes through the input data improves the quality of the generated clusters but also extends the processing time required to perform clustering. We restricted the number of passes to 20 (more then 10 passes are preferred) and the number of clusters to be generated up to 16 in order to find accurate and homogenous groups of objects and simultaneously avoiding the production of too many small clusters. As the active fields (attributes), i.e. record fields that participate in creation of clusters, we chose: DAY, HOUR, HOP and the average RTT. The results of the clustering function show the number of detected clusters and the characteristics that make up each cluster. We obtained the following mining outputs: Number of Passes Performed: 20, Number of Clusters: 16, Clustering Deviation: 0.01082. The results can be visualized using graphical characteristics for all clusters and for each individual one. The results are normalized. This information can be analyzed by the user now or better after predicting a cluster using the decision tree. E.g. the user can perform the sensitivity analysis using the sensitivity report which is generated as a list of input fields ranked according to their respective importance to the classification function.

First eleven clusters (76% of total population) are almost similar in their sizes, each size from a range 6-8 % of the total population. General cluster description of the six top clusters is shown in Fig. 1. The clusters differ from each other. For instance, cluster 14 (7.93% of population) defines the set of records where RTT is high, DAY is predominantly 3, HOP is predominantly 19 and HOUR is predominantly 16. For such cluster description the modal values for the relevant fields are respectively: 200-225 ms, 3, 19, 16.

Id	Relative cluster size (%)	Cluster Description
14	7.93	RTT is high, DAY is predominantly 3, HOP is predominantly 19 and HOUR is predominantly 16
3	7.79	DAY is predominantly 1, HOUR is predominantly 2, HOP is predominantly 20 and RTT is medium
7	7.69	DAY is predominantly 1, HOUR is predominantly 9, HOP is predominantly 20 and RTT is medium
12	7.05	DAY is predominantly 7, HOUR is predominantly 23, RTT is medium and HOP is predominantly 20
8	6.97	DAY is predominantly 7, HOUR is predominantly 15, RTT is medium and HOP is predominantly 20
4	6.91	DAY is predominantly 7, HOUR is predominantly 11, RTT is medium and HOP is predominantly 20

Fig. 1. Description of the six top clusters

The next (according to its size) cluster 3 (7.79%) defines the set of records where DAY is predominantly 1, HOUR is predominantly 2, HOP is predominantly 20 and RTT is medium. Then modal values for the relevant fields are respectively: 1, 2, 20 and 50-75 ms. The order of active fields in the definition of cluster shows the importance of the field. The field distributions within the clusters tend to be different from their global distributions.

We can analyze cluster details, for example, of cluster 14. The records assigned to the cluster 14 represent the measurements, which in most (about 42% of records belonging to that cluster) situations are performed on Wednesday (DAY=3) or on Thursday (39%), with the number of hops predominantly 19 (65%). These measurements are performed mostly between 12:00 and 23:00 (98%) with the predominant hour 16:00 (12%). The other 2% of records include remaining twelve values that might appear in the HOUR field. In 14% of measurements represented in that cluster there are round-trip times from the 200-225 ms range. Each of the RTT ranges 175-200, 275-300, 300-325 ms includes 12% of records. RTT less than 100 ms or greater than 425 ms never occurred. There are mostly 19 hops on the path between our hosts (65%). 20 hops are identified in 31% measurements whereas only 4% of paths have 19 hops. In comparison, the total population has different distribution: RTT is predominantly medium (i.e. 50-75 ms, 39% of total population), DAY and HOUR have even distributions and HOP is predominantly 20.

4 Application of Classification Mining Function

One of the disadvantages of cluster models is that there are no explicit rules to define each cluster. The model is thus difficult to implement, and there is no clear understanding of how the model assigns cluster IDs. Therefore we need the next step in building our model. We propose to use the classification mining function. Classification is the process of automatic creation of a model of classes from a set of records. The induced model consists of patterns, essentially generalizations over the records that are useful in distinguishing the classes. Once a model is induced, it can be used to automatically predict the class of other unclassified records. Decision tree represents the knowledge in the form of IF-THEN rules. Each rule can be created for each path from the root to a leaf. The leaf node holds the class prediction.

IBM Intelligent Miner for Data offers tree induction (modified CART regression tree) or neural networks algorithms (back propagation) to compute the classes [18]. Like neural networks used previously, the trees develop arbitrary accuracy and use validation data sets to avoid spurious detail. Unlike neural networks, trees are easy to understand and modify. In many instances the decision tree produces a very accurate representation of the cluster model (>90% accuracy). If the tree representation is accurate, it is preferable to implement a tree, because it provides explicit, easy-to-understand rules for each cluster. That is why we used a decision tree to classify the cluster IDs using the output data that was obtained as a result of applying the clustering algorithm. When building the tree, we did not limit the number of node levels for the binary tree to be created, but we assumed that 5 records were included in an internal node before it split to the next node. The general objective of creating the model is to use it to predict RTT values and the number of hops most probable to achieve in the future. We assume that the only a priori information that is to be given in the future is the day of the week and the hour of the day. These fields were specified as active fields participating in classification mining function. First we use our mining system in the training mode to build a model based on the selected input data. This model is later used as classifier. After then we tested the model on the same data with known class values to verify that the model created in training mode produces results of satisfying precision. Finally in application mode, we used a model created in training mode to predict the specified fields for every record in the new input data (measurements from another month). We split the whole data set into training and testing data. The measurements from last two weeks were used as the testing data whereas the remaining measurements were used for training. The tree classifier built in training mode the tree with 47 nodes and depth 12. This tree showed 91% of correct classifications while testing. Fig. 2 shows the decision tree rebuilt after the pruning of some nodes (31 nodes were pruned for prune level 4). Pruned tree has worse number of correct classifications, namely 79% but is simpler. The purity in leaf node indicates the percentage of correctly predicted records in that node.

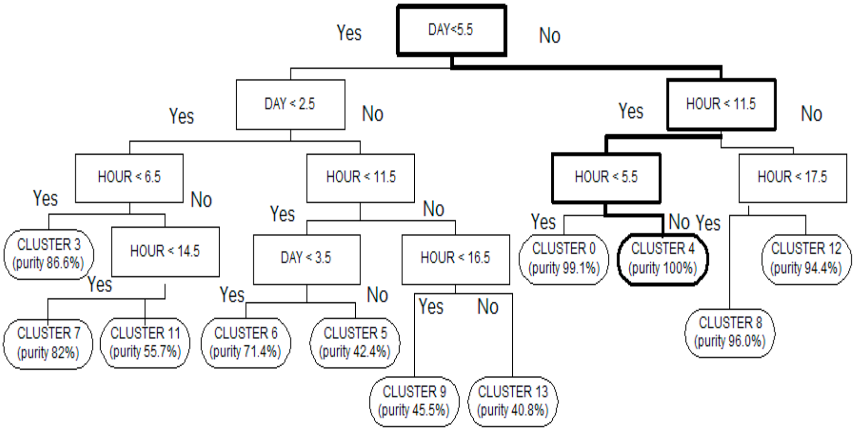


Fig. 2. Decision tree model built by the classification function (after pruning)

The decision tree consists of rules, which can be used by the user in forecasting the path performance. The user can take advantage of this knowledge, e.g. for scheduling transfer of data in that period of time. For example, the following rule for cluster 4 can be applied when using the pruned tree: if our Internet activity (on the path under consideration) is to be scheduled on Saturday or Sunday (DAY=>5.5) and between 5 o'clock (HOUR=>5.5) and 11:30 (HOUR<11.5), than we can suppose that the network performance will be similar to the conditions described by cluster 4, that is the RTT will be medium (i.e. 50-75 ms) and HOP will be predominantly 20.

5 Conclusions

This paper presents some lessons from the application of data mining in a new application area. We have mined the RTT measurements collected over specified period of time. We showed how information about past performance experience can help users identify potential performance problems in the future and decide how to use the Internet paths. We demonstrated how two data mining functions, namely clustering and classifying, can be used to discover patterns in performance of end-to-end Internet path. The decision tree model was built and tested using sample data. It showed good accuracy of 91% and can be used to predict Internet path performance. The model created was the best at the moment and for the specific Internet path. It should be continuously updated when new measurements are performed. There is the need to develop new algorithms for doing incremental data mining for such case. Although we dealt here only with data mining used for characterization of a single host-to-host path we do hope that data mining approach can be used in much advanced investigations on Internet behavior.

References

1. Borzemski L., Nowak Z.: Estimation of HTTP throughput and TCP Round-Trip Times, Proc. of 10th Polish Teletraffic Symp., IEEE Chapter, Cracow (2003) 335-352
2. Barford P., Bestavros A., Byers J., Crovella M.: On the Marginal Utility of Network Topology Measurements. In: Internet Measurement Workshop, San Francisco, (2001) 5-17
3. Brownlee N., Claffy K, Murray M., Nemeth E.: Methodology for Passive Analysis of a University Internet Link. In: Proc. of Workshop on Passive and Active Measurements PAM2001, Amsterdam, Holland (2001)
4. Brownlee N., Loosley C.: Fundamentals of Internet Measurement: A Tutorial. Keynote Systems, USA (2001)
5. Cardellini V., Casalicchio E., Colajanni M., Yu P.S.: The State of the Art in Locally Distributed Web-Server Systems. ACM Comp. Surveys, Vol. 34, No. 2, June (2002) 263-311
6. Claffy K., McCreary S.: Internet Measurement and Data Analysis: Passive and Active Measurement. University of California, CAIDA, USA (1999)
7. Cross-Industry Working Team (XIWT). Customer View of Internet. Service Performance: Measurement Methodology and Metrics. (1998)

8 Cross-Industry Working Team (XIWT). Internet Service Performance: Data Analysis and Visualization. (2000)

9 Faloutsos M., Faloutsos Ch.: Data-Mining the Internet: What We Know, What We Don't, and How We Can Learn More. ACM SIGCOMM 2002 Conference, Pittsburgh, PA (2002)

10. Garofalakis M., Rastogi R.: Data Mining Meets Network Management: The NEMESIS Project., Proc. of DMKD'2001, Santa Barbara, California (2001)
11. Grossman R. L., Kamath Ch., Kegelmeyer P., Kumar V., Namburu R. R. (Eds.): Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, Boston, Dordrecht London (2001)
12. Hellerstein J., Zhang F., Shahabuddin P.: A Statistical Approach to Predictive Detection. Computer Networks 35 (2001) 77-95
13. Luckie M. J., McGregor A. J., and Braun H.-W.: Towards Improving Packet Probing Techniques. In: Internet Measurement Workshop, San Francisco, CA (2001) 145-151
14. Murray, M. claffy, k.: Measuring the Immeasurable: Global Internet Measurement Infrastructure. In: Proc. of Workshop on Passive and Active Measurements PAM2001, Amsterdam, Holland (2001) 159-167
15. Ng E. T. S., Zhang H.: Towards Global Network Positioning. In: ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA (2001) 25-29
16. Saroiu S., Gummadi K. P., Dunn R. J., Gribble S. D., Levy H. M.: An Analysis of Internet Content Delivery System. In: Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA (2002) 315-327
17. Srikant R., Yang Y.: Mining Web Logs to Improve Website Organization. In: Proc. of WWW10 Conference, Hong Kong (2001) 430-437
18. Using Intelligent Miner for Data. V8 Rel. 1, IBM Redbooks, SH12-6394-00 (2002)
19. Wolski R. Dynamically Forecasting Network Performance Using the Network Weather Service. Technical Report TR-CS96-494, U.C. San Diego, CA (1996)
20. Zhang Y., Duffield N., Paxson V., Shenker S. On the Constancy of Internet Path Properties. In: Internet Measurement Workshop, San Francisco, CA (2001) 197-211
21. http://www-iepm.slac.stanford.edu, http://www-iepm.slac.stanford.edu/pinger/

Change Summarization in Web Collections

Adam Jatowt, Khoo Khyou Bun, and Mitsuru Ishizuka

University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan
{jatowt, kbkhoo, ishizuka}@miv.t.u-tokyo.ac.jp

Abstract. World Wide Web is not only enormous but also dynamic information space. Every day large quantity of new information is published on web pages. Many times people want to know what are the major changes in their area of interest over a given time period. This paper addresses the problem of summarizing changes in web collections devoted to a common topic. We have created a system called ChangeSummarizer, which periodically monitors a web collection in search for new changes and generates their summary. Since many web pages can be quite static over long time or just unrelated to the query we employ the method to evaluate, which pages are dynamic and which provide valuable content. Basing on this evaluation ChangeSummarizer creates web page ranking list and updates it regularly in order to improve subsequent summaries. Additionally, the system searches for new, valuable web pages, which can be included into the collection to enhance its quality.

1 Introduction

Internet has become the biggest information repository in the world. The large quantity of available data and the dynamic nature of web pages make information retrieval a challenging task. User interested in a certain topic can exploit many information resources of various nature, content and characteristics. The low cost of publishing information in WWW pages results in an unpredictable and quick changes of document contents. User can be overwhelmed with the quantity of news sources and may not be aware, which of them contain valuable and up-to-date information. Our system assists users in searching for new relevant information by providing them with the summary of recent, important changes related to specified topic.

We would also like to introduce a new research area called “change summarization” or more specifically “multi-document change summarization”. The idea is to collect textual changes in related documents over certain time interval and to produce their summary. Such summary would ideally display the most important, popular changes occurring in the whole collection of web documents. There are several situations when change summarization can be of some value. Users may want to know the most important changes occurring in some domains. They can be interested in popular topics discussed in their area of interest or for example in the changes in opinions of web page authors during a specified period. Web pages in contradistinction to standard documents can change their contents unlimited number of times. We can say that changes reflect the dynamic character of a document. Generally, a web page should be considered as a dynamic document or as an

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 653-662, 2004.

information slot where a new content can be placed in undefined time. However, it is difficult to predict the scope and time of web page changes. Usually for newswire sources one can be quite sure that fresh news will be published on a daily or weekly basis. Nevertheless, the situation can be different in case of other types of pages. The changing text of the page can have various sizes. The most extreme case happens when the whole document is deleted or a new one is created. In other cases some new information is inserted or old text is deleted in addition to some unchanged, static context. The textual changes can have various meanings however we assume that, to high extent, they are topically related to the old versions of the page and to the entire collection.

Each WWW page from the collection is periodically checked for new textual data. After comparison of new and old versions of all pages from the set, the most important terms are extracted. The system calculates scores for each term according to the popularity of the term in static and dynamic parts of the collection. Basing on the ranking list of important terms occurring within the examined period, we select sentences with the highest overall scores and present them to user. Apart from these methods we decided also to exploit the knowledge hidden in the history of each web document activity. The notion of "up-to-date-ness" function for a given web page is introduced in order to evaluate how often textual changes appear on the page and how topically close they are in comparison to the other documents. This function is based on web page dynamic scores, which are calculated every time the WWW page is examined for new changes. They specify the significance of new textual data found at this page with regards to dynamic content of other pages from the collection. ChangeSummarizer maintains a ranking list of the web pages constituting collection, which is updated after each change monitoring. Therefore web pages, which are not changing frequently or whose changes do not contain sufficiently enough popular terms will not be taken into account during generation of next summaries. Furthermore, the top 15 pages from the ranked list are periodically exploited as the base for discovery of new documents. The system searches for other related or similar web documents. After inclusion into to the collection these new sites have dynamic scores assigned as in case of the rest of pages constituting the set. In this way we aim at creating and maintaining the collection of relevant to the topic and frequently updated web pages.

In the remainder of this paper, we describe our efforts towards summarizing changes in online resources. In the next section we review some related systems and solutions. Section 3 discusses the architecture of ChangeSummarizer. In Section 4 results of our experiment are presented. We conclude in the last section and outline future research directions.

2 Related Work

In the last years several attempts have been made to design systems for developing automatic reports of important events [5], [3], [6]. Usually these applications search for popular and important topics basing on predetermined set of web pages. Google News [3] tracks several thousands of news sources. User can issue a query and read related, recent articles. However Google News does not produce typical summaries but rather displays links to variety of sources discussing any given event. The other systems like Newsinessence [6] or Newsblaster [5] provide summaries of popular recent events, which are discussed in some chosen news sources. However, there is a need for an application that could summarize information from any types of web pages. In other words it should be a system that could produce summaries of collections of web pages, which are not limited to newswire extracts. WebInEssence [7] is an example of such application, attempting to generate summaries from dynamically constructed groups of web pages. ChangeSummarizer, on the other hand, summarizes textual changes in web collections. We focus on new, changed data in various kinds of web documents searching for common information. Additionally, our system expands web collection by searching for related web pages.

There are several systems designed for detecting and visualizing changes in WWW pages. Any user-specified features like links, text, pictures and etc. can be tracked. Usually change detection applications require a user to provide web page address to be monitored [4], [1]. The results can be sent by email as a list of changes or as a composition of different page versions for better visualization of changes. However, user is often overloaded with meaningless changes like, for example, modified syntax or color. In spite of this drawback, there was little research done on the extraction and summarization of meaningful changes from web pages.

3 ChangeSummarizer System

The conceptual architecture of our system is displayed in Figure 1. It shows the information flow in one cycle of ChangeSummarizer's performance, which takes place between downloading two consecutive versions of web collection. Each such phase is executed periodically within pre-defined time interval. Changes are examined throughout the whole tracking process, which contains some number of singular phases. The longer time the system is running, the more information can be gathered and later used as a collection history data.

The time interval between two consecutive downloads of web collection versions has an impact on overall "recall of changes" and on a single page influence on the summary. The shorter this period, the more efficient the system is in detecting short-life changes. However, usually such brief time is not sufficient enough for many web pages to change. Therefore only one or few of them will have any changes at all. In result it may happen that a single document change has relatively high impact on the final summary. The scope of this influence depends on the character of web pages constituting the collection. On the other hand, in case of the longer period one should obtain more changes, which in the end will diminish the influence of a singular web page on the summarization result. However there is a risk of losing some dynamic data during this extended time. Simply some documents may change their contents more than once during the period. In consequence, the "recall of changes" can be lower especially in the case of fast-changing resources.

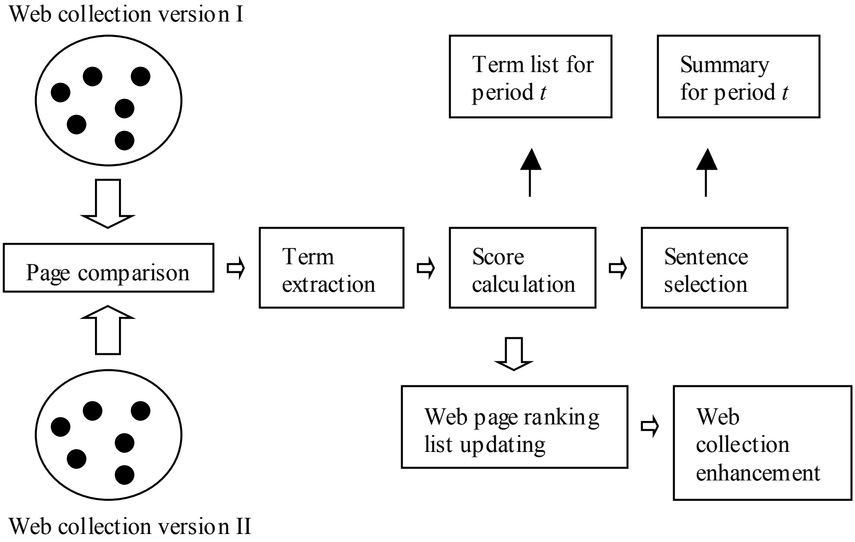


Fig. 1. Information flow in the system

3.1 Change Detection and Term Scoring

We can collect web pages relevant to user's interest in two ways: by issuing query to search engine or by using some popular web directories like, for example, Yahoo! In case of the first way, the decision of the right query words is very important since tracking results will naturally depend on the content of a web collection. Elementary base set of web pages is created by fetching first 200 hits generated by search engine in response to the user query. ChangeSummarizer checks also for duplicate pages, which must be discarded. On the other hand, the second method is a straightforward one since it utilizes already clustered and human-edited set of web documents.

Periodically or according to an arbitrarily specified time schedule, ChangeSummarizer extracts changes from the web collection by comparing old and new versions of each web page. User can specify the threshold or percentage of web pages, which will be considered for summary. We have decided to process 75% of web pages with the highest scores in the ranking list of the collection.

Next, textual data must be extracted from downloaded HTML files and converted into plain text format. Retrieval of new information (changes) is conducted by the comparison of sentences from the consecutive document versions. If a given sentence from the latest version does not appear in the previous version of the web page, then it is regarded as a new one.

In the following step, ChangeSummarizer separates words from the changed sentences and subjects them to stemming. To eliminate semantically poor words the system conducts basic stop-list filtering. Regarding the selection of features for summary creation, we have decided to use *n*-grams, where *n* is from *1* to *3*. *N*-grams are combinations of consecutive *n* words. ChangeSummarizer calculates *n*-grams for changed parts during each cycle and also eliminates low frequency terms to reduce the feature dimension. *N*-grams are treated as separated entities in an equal way as single words.

Consequently, a weight S_i, denoting a score of "popularity" for a term *i*, is computed using the following weighting scheme.

$$S_i = \left(1 + \frac{\sum_{j=1}^{N_{doc}} \left[\frac{n_{jc}}{N_{jc}+1} - \alpha * \frac{n_{js}}{N_{js}+1} \right]}{N_{doc}} \right) * \exp\left(\frac{n_{icp}}{N_{cdoc}+1} - \alpha * \frac{n_{isp}}{N_{sdoc}+1} \right). \qquad (1)$$

Table 1. Explanation of symbols used in Equation 1

S_i - *score for term i*	n_{isp} - *number of pages where static parts contain term i*
N_{doc} - *number of pages in the collection*	n_{icp} - *number of pages where changed parts have term i*
N_{js} - *number of static terms in page j*	N_{jc} - *number of changed terms in page j*
N_{sdoc} - *number of static documents*	n_{jc} - *number of term i in changed part of page j*
N_{cdoc} - *number of changed pages*	n_{js} - *number of term i in static part of page j*

Table 1 explains the meanings of individual symbols used in Equation 1. This equation defines the score of a term as its "popularity" in changes in the web collection. However, additionally, the score is also influenced by term's "unpopularity" throughout static parts of current web collection snapshot. Consequently, terms with high scores should appear often in changed parts of many web pages but rarely in static parts of documents. In this way, user may find not only popular terms in changes but also terms, which are unexpected since they do not occur frequently in static parts of documents. Such unexpected terms can be interesting to a user who is an expert in his area. Another motivation for this approach is that terms appearing frequently in changes may have low semantic values for a given topic. Since we want the system to be domain independent, we employ only one general stop-list. However each topic has different terms that are considered as semantically poor in given domain. Therefore we should assign higher scores to terms that do not occur frequently or are not typical words of specific domain chosen by user. Parameter α is used to specify the relative weight of such "unexpected" terms. Its range is from *0* to *1*, where the value equal to *0* indicates that there is no influence of the static part of collection on the term selection. Common terms in changed (dynamic) parts of documents are represented as the exponent of frequency of a term in those parts of web pages. For better explanation we can conceptually divide the Equation 1 into two parts. The first one describes how often a given term occurs inside each document on average. This part can have values between *0* and *2* depending on the term distribution in every single document and on parameter α. The second part is the exponent of the term distribution among all documents in the collection. It has major impact on overall score since terms popular in many changed parts of documents should reflect common changes.

3.2 Page Up-to-date-ness and Web Collection Enhancement

Some web pages change within high frequency and have meaningful contents related to the query. Including such web documents into the collection could improve the quality of subsequent summaries. Thus it would be beneficial to select and utilize them to greater extent. In order to do so, the system creates web page ranking list. The rank of a page is based on singular dynamic scores S_t, which are assigned to the web page for every period t of consecutive cycles of the system. As was said before, periodically, we obtain a list of common weighted terms taken from all changes. Thus for a given page it is possible to describe the value or "commonness" of its dynamic content simply by summing weights of all terms and dividing the acquired sum by the number of all terms in this piece of text N_d (Equation 2).

$$D = \sum_{t=1}^{T} \frac{S_t}{(T - t + 1)} \quad ; \qquad S_t = \frac{\sum_{j=1}^{N_d} S_{jd}}{N_d} \, . \tag{2}$$

We also consider the frequency of web page changes as another type of measure of web document value. Thus we need to take into consideration the number of times that the web page changed during the whole monitoring process. Our idea is to sum the dynamic scores of the web page with respect to their dates of occurrence. The latest additions of new information indicate higher usefulness of the web page in the present moment. Therefore, the latest changes should be scored higher then changes, which occurred somewhere in the beginning or in the middle of page tracking. We are proposing an up-to-date-ness function D of a web page expressed as a sum of singular dynamic scores, whose value is decreasing along with time (Equation 2). Variable T denotes the number of time units, which passed from the beginning of the change tracking process. Web pages are sorted in the ranking list according to the value of function D. Figure 2 shows an example of the up-to-date-ness function (thick line) where the singular dynamic scores S_t of the page are represented as the starting points of thin lines.

Except for change extraction and summarization, new resources should be constantly discovered and included into the collection to enhance its quality. After the inclusion of new web pages, the scores would be computed to calculate their positions in the ranking list. Basing on this approach the consecutive summaries should be more correct and up-to-date.

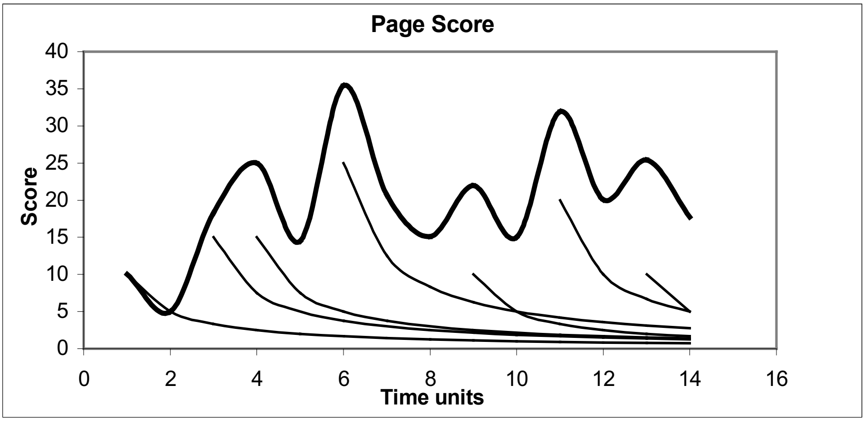


Fig. 2. Example of up-to-date-ness function

In order to automatically discover new informative web pages, ChangeSummarizer searches for web sites related to the few high-scored documents from the ranking list. Our method is a modification of Co-citation algorithm [2] in the sense that we search for resources related to several web pages simultaneously rather than only to one web page. Usually 15 web pages with highest ranks in the list are examined for their in-links. It means that the system looks for web pages which link to this selected group of documents. Then common in-links are arranged according to their frequency and 200 of the top parent pages are downloaded and stored in a separate folder. These are the documents that have the highest number of links leading to different pages from the group of 15 best resources. As a next step, the system investigates out-links from these retrieved 200 web pages searching for the frequent ones. Finally, the most common 10 links are selected. Then the web pages that are pointed to by these links are fetched and included into the collection as new resources. Consecutive change tracking phases will process these pages and use for summarization. It is important to state that only original pages should be inserted into the collection. Duplicate ones would not only influence the results but also would cause the miscounting of links during next collection updating phases. Therefore we filter duplicate pages, as well as, pages from the same sites, that means, web documents, which have identical domain names. This is due to the fact that pages within the same site may contain identical link set, which influences link counting.

3.3 Sentence Selection

In the result of change tracking, the system stores the ranked list of the most common terms together with their responding weights. To produce the final summary one needs to select representative sentences from the content of changes. ChangeSummarizer calculates the overall weight for each new sentence to pick up the ones that convey the meaning of main changes in a given period. Consequently the user is presented with the list of few most highly weighted sentences for each cycle of the system. To increase readability we add also preceding and following sentences surrounding selected top sentences. Additionally, such extracts contain links to their host web pages to enable the user to read the whole document if necessary. Sentence scoring formula is illustrated in Equation 3.

$$S_{sen} = \frac{\sum_{i=1}^{N_{sen}} S_i}{N_{sen}} * \left(1 + \beta * \frac{N_{doc} - S_{page}}{N_{doc}}\right). \tag{3}$$

Score for a sentence is basically the sum of the scores of its all terms divided by the number of these terms. Furthermore, we make use of the historical data, which has been acquired during earlier change tracking phases. Thus we modify the sentence score by the page weight of the document, where a given sentence was published. The symbol S_{page} indicates the web page position in the ranking list of the collection. Parameter β corresponds to the strength of the "historical data" influence and has a range from *0* to *1*.

As was mentioned before, 75% of the highest scored web pages are utilized for summarization. Nevertheless, ChangeSummarizer continually computes scores for the remaining pages in the lower positions of the list. Thanks to that, there is a chance to include them for forthcoming summaries provided that their characteristics will improve.

4 Results

We present the results from the experiment conducted for the query "latest movies". The collection was obtained after issuing the query to a search engine and fetching the top 200 web pages. Change tracking was performed several times during period from 18th February to 12th May. Table 2 displays top-scored terms obtained during 4-days interval from 26th to 29th April 2003. Due to limited space we present only the top-scored terms for one time interval.

On the 25th April two popular movies were released such as "It runs in the family" and "Identity". The first one is about the life of three generations of New York's family with Kirk and Michael Douglas starring. The later one shows the sequence of mysterious murders committed in motel during one night. The main actor in this movie is John Cusack.

We show three sets of highest-scored terms for different values of parameter α. Along with the increasing value of α terms having rather general meaning such as: "movie", "film" or "make" are descending to lower ranks in the list. On the other hand rare or specific to the above movies terms like: "douglas", "motel" and "identity" have higher relative scores.

Table 3 displays top sentences for different time intervals for parameter α value equal to *1*. Basically they provide information or comments on new movies, which have been or are going to be released in the nearest time. Given the diversity of types of pages and topics related to the query, it should not be surprising that the final results may not constitute coherent summary. Intuitively, it is very important to construct appropriate web collection with closely related web pages. We have noticed that the system produces better results for narrow topics, where documents tend to be

Table 2. Top terms for different parameters α

Parameter α = 0		**Parameter α = 0.5**		**Parameter α = 1**	
movi	1.649009	star	1.395895	star	1.306643
new	1.564911	new	1.354146	man	1.247069
star	1.485302	movie	1.320918	family	1.245389
film	1.435972	like	1.315999	douglas	1.23617
like	1.412125	man	1.293055	like	1.21999
just	1.362831	family	1.280962	girl	1.215398
time	1.339155	girl	1.265839	run	1.207945
man	1.339055	just	1.25945	murder	1.172834
family	1.316553	film	1.255065	college	1.166521
want	1.316437	douglas	1.243223	dark	1.165882
girl	1.316373	run	1.239737	start	1.165546
make	1.295729	year	1.225889	motel	1.160303
look	1.294302	want	1.209694	year	1.158099
year	1.293781	start	1.208024	just	1.156253
run	1.271534	make	1.20319	cusack	1.14825
start	1.250516	murder	1.190517	John cusack	1.148215
play	1.250359	dark	1.187093	identity	1.14806

Table 3. Final sentences

Top sentences with following and preceding sentences	**Period**
"It was the first time three generations of one family have been in a picture," says the elder Douglas. The father and son star in the movie, along with Michael's son Cameron, 24, and Michael's mother, Diana Douglas, long divorced from Kirk. Sitting in the elder Douglas' elegantly appointed one-story home, Kirk, 86, and Michael, 59, clearly have a warm relationship.	26/4-29/4
The result is "Ghosts of the Abyss," an hour-long triumph of documentary filmmaking and a new high-mark in the director's already fabled career. Rock star Rob Zombie makes his debut as a feature film writer and director with "House of 1000 Corpses," opening this weekend at Cinema World in West Melbourne. Zombie's only previous feature experience was an animated sequence in "Beavis & Butt-head Do America," though he has directed several music videos.	4/4 - 12/4
James Cameron mixes CG and 3D. Dark Horizons reports that the Titanic director's next film will be in the same vein as Avatar, which was going to be the first film with total CG actors in it. He said it wouldn't be as big scale as that, but would have some CG characters in it.	16/4-26/4

more topically related. In the collection except for several newswire resources or web sites solely devoted to topic, there have been found also some message, opinion boards and "blog" type pages. Although, such pages can blur the final results, they provide information about opinions or hot topics in contradistinction to information about rather pure facts presented by newswire sources.

5 Conclusions and Future Work

We have introduced a new research area of summarizing textual changes in web page collections and have presented a complete, cyclical system called ChangeSummarizer. The system uses novel methodology for extracting and summarizing textual changes in web collections. Summarizing changes is based on searching for common and semantically rich content terms. ChangeSummarizer maintains ranking list of web pages, where each page is scored according to the frequency and the contents of its changes. In this way historical data can be gathered and later exploited for the summarization purposes. The most valuable web pages, according to this measure, are utilized for consecutive summaries. Additionally they form a base for finding new web pages to be included into the collection.

Our system has several limitations, which we want to focus on in the future. One problem concerns changes in the form of old and new links found in web documents. If web pages linked by a certain document are found on the same web site as this document then we should also consider their changes. Moreover, since new sentences can be placed in different semantic contexts on a web page, one should take into consideration not only the content of a changed sentence but also its relation to the surrounding text.

References

1. Changedetect: http://www.changedetect.com/
2. Dean, J., Henzinger, M.: Finding Related Pages in the World Wide Web. In Proceedings of The Eighth International World Wide Web Conference. Toronto Canada (1999) 1467-1479
3. Google News: http://news.google.com
4. Liu, L., Pu, C., Wang, T.: WebCQ: Detecting and Delivering Information Changes on the Web. In Proceedings of International Conference on Information and Knowledge Management. Washington, DC, USA (2000) 512-519
5. McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J.L., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S.: Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In Proceedings of Human Language Technology Conference. San Diego, USA (2002)
6. Radev, D., Blair-Goldensohn, S., Zhang, Z., Raghavan, S.R.: NewsInEssence: A System for Domain-Independent, Real-Time News Clustering and Multi-Document Summarization. In Human Language Technology Conference. San Diego, USA (2001a)
7. Radev, D., Fan, W., Zhang, Z.: WebInEssence: A Personalized Web-Based Multi-Document Summarization and Recommendation System. In NAACL 2001 Workshop on Automatic summarization. Pittsburgh, USA (2001b) 79-88

Control System Design for Internet-Enabled Arm Robots

S.H. Yang* [1], X. Zuo[2], and L. Yang[3]

[1] Computer Science Department, Loughborough University,
Loughborough, Leicestershire LE11 3TU, UK
s.h.yang@lboro.ac.uk

[2]Automation Research Institute, Petroleum University, Beijing, 102200, PR China
zuoxin@bjpeu.edu.cn

[3] Computing Department, Business School, University of Derby, Derby, DE22 1GB, UK
l.yang@derby.ac.uk

Abstract. Internet-based robotic systems have received much attention in the recent years. A number of design issues are essential for designing this new type of robotic systems. This paper addresses the user interface design and control structure selection for general Internet-enabled robots. An Internet based control system for an arm robot has been used as a case study to illustrate these general principles, in which a multimedia based user interface is built and an open-loop control structure is implemented.

1 Introduction

In the last decade, the most successful network developed has been the Internet. It has made a significant impact on society through its use as a communication and data transfer mechanism. Many systems are being created all over the world to implement Internet applications. Most of them are being focused on tele-robotic systems [2, 4, 6, 7]. The creation of virtual laboratories for education purpose is also one of the areas that are being currently developed [3, 8]. In the area of control systems some work has been done for guiding the design process, dealing with Internet latency, and assuring the safety and security [9, 10, 12].

Internet-based robotic systems have received much attention recently. When we see the Internet as an infrastructure on which to build a robotic system, its attraction is three-fold. First, web browsers can provide a nice human interface for a robotic system, because the browsers can display various media including hypertext, moving images, sounds, and three-dimensional graphics as well as handling interactive operations of the media. Second, hypertext transfer protocol (HTTP) can be a standard communication protocol of a tele-robotic system, since robots connected to the Internet can be accessed from any Internet site via the protocol. Third, it becomes possible to use various robotic hardware/software distributed over the Internet together to accomplish a single mission.

* Corresponding author

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 663-672, 2004.

Therefore, it is natural to consider developing robotic systems on the Internet. Many robotic systems have been developed on the web, despite that the latency of the Internet is unpredictable. The tele-operation system, “TELEROBOT”, at the University of Western Australia allows the Web user to control a robot arm [7]. Another example is the Bradford Robotic Tele-scope [2], through which the WWW users can look at an image taken from an observation with the telescope and compare it with one taken from a star database held at NASA.

This paper describes our experiences in building an interactive web interface and selecting control structures for operating and managing an Internet-enabled arm robot. This paper is structured as follows. Section 2 presents the challenges in the design of Internet based control systems. Control structure selection is presented in Section 3. A multimedia user interface is illustrated in Section 4. An open-loop control structure is selected in Section 5. The implementation of the Internet based control system for our arm robot is introduced in Section 6. Finally Section 7 concludes this paper.

2 Challenges of Internet-Based Control

Introducing the Internet into control systems has introduced a number of challenges. Two of them are discussed in this section.

2.1 Internet Transmission Delay

The transmission latency is the main difference between Internet-based control and other tele-operation. Most tele-operating systems are based on private media, by which the transmission delay can be well modeled. The Internet in contrast is a public and shared resource in which various end users transmit data via the network simultaneously. The route for transmission between two end points in a wide area is not fixed for different trails and the traffic jam may be caused when too many users traverse the same route simultaneously. The transmission latency of the Internet is difficult to model and predict. The reasons why the variable time delay occurs are as follows:

- Network traffic changes all the time because multiple users share the same computer network.
- Routes or paths of data transmission decided by Internet Protocol (IP) are not certain. Data is delivered through different paths, gateways, and networks whose distances vary.
- Large data is separated into smaller units such as packets. Moreover, data may also be compressed and extracted before sending and after receiving.
- Using TCP/IP protocols, when error in data transmission occurs, data will be retransmitted until the correct data is received.

To reduce the effect of the time delay, most systems try to extrapolate forward environmental information and manipulator states in time. Some works use local simulated manipulators or simulators to assist in controlling the remote devices. Selecting a proper control structure is also a promising way to overcome the time delay.

2.2 Web User Interface Design

The central design objective for a web-based user interface in Internet-based control is to enable the operator to appreciate more rapidly what is happening in the robots and to provide a more stimulating problem-solving environment outside the central control room. It should be borne in mind that media available in the Internet environment outside the central control room will be very much limited compared to those in the central control room.

The technologies from the areas of "multimedia" and "Virtual Reality" show considerable potential for improving yet further the human-computer interfaces used in control technology [5], and different media can transmit certain types of information more effectively than others and hence, if carefully chosen, can improve operator performance [1]. Fig. 1 illustrates the features of various media. Choosing the best media for different interface tasks and minimizing the amount of irrelevant information in the interface are two main guidelines in the user interface design.

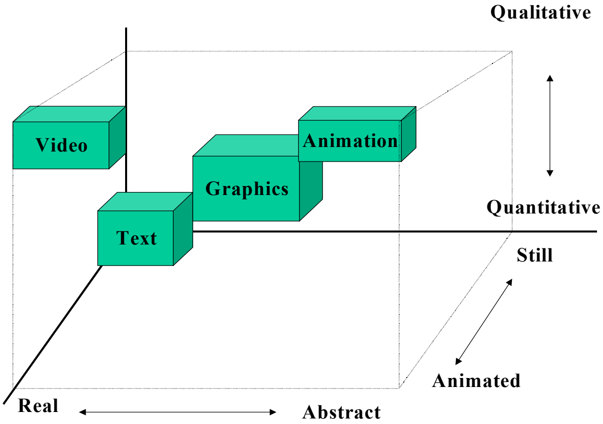


Fig. 1. Web user interface design

3 Possible Control Structures for Internet-Enabled Robots

The traditional control system structure is shown in Fig. 2. An operator gives a desired input to the controller. The controller outputs the control signal to the actuator based on the difference between the desired input and the measured output. The actuator passes the control action to the robot and regulates its behavior. The measured output is fed back to the controller through the sensor. In this section three possible control structures are summarized. The pros and cons of each structure are also analyzed.

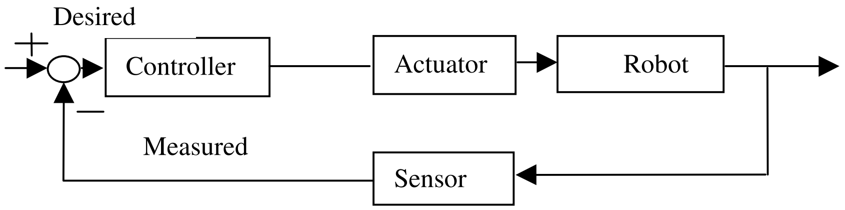


Fig. 2. Traditional control system

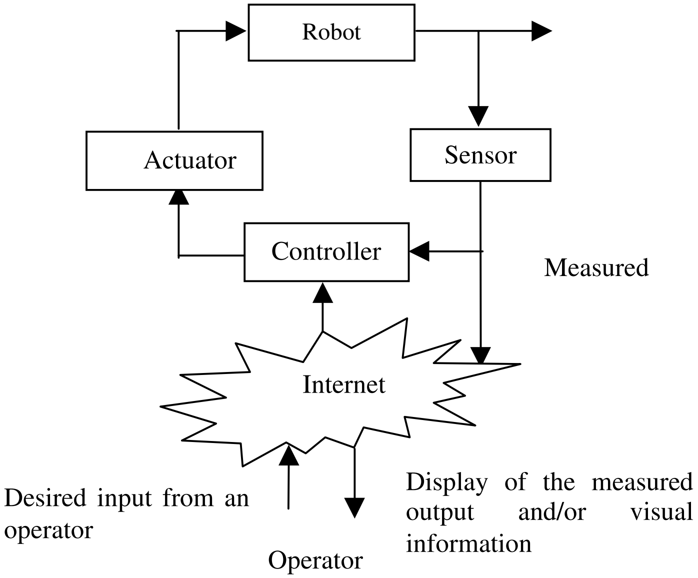

Robot
Actuator
Sensor
Controller
Measured
Internet
Desired input from an operator
Display of the measured output and/or visual information
Operator

Fig. 3. Control structure with the operator located remotely

The straightforward control structure over the Internet is to allow the operator located in the remote site to send control commands (desired input) to the controller located in the local site with the robot through the Internet. The structure is shown in Fig. 3. In order to monitor both the performance of the controller and the situation of the robot the measured output and/or some visual information are required to feedback to the operator at the remote site. Because the Internet is excluded from the closed loop and the controller is located at the same location with the robot the Internet transmission delay will not affect the performance of the control system. Obviously the Internet transmission delay will affect the transfer of the desired input from the operator site to the controller site. Some measures must take place to compensate these effects.

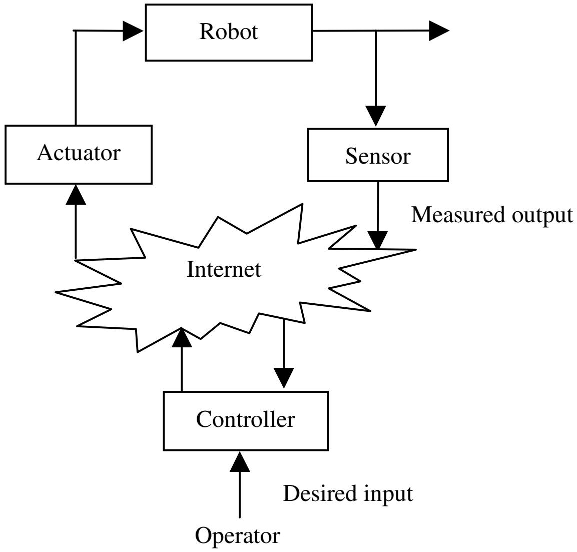


Fig. 4. Control structure with the controller located remotely

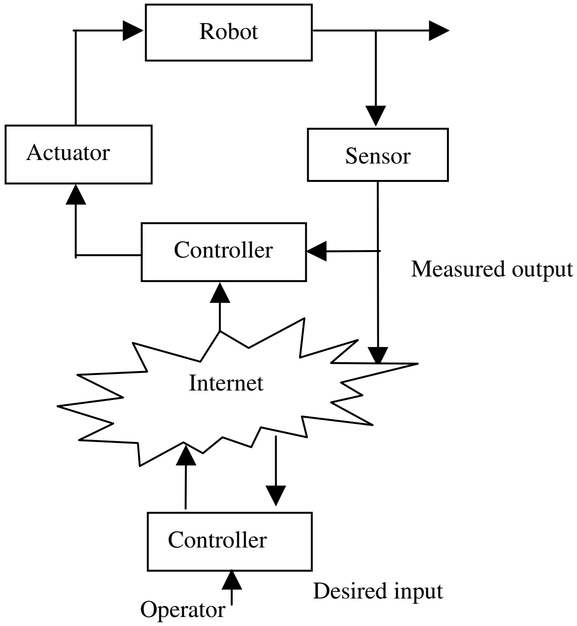


Fig. 5. Controı structure with bilateral controllers

In some cases such as virtual control laboratory and remote design of controllers, it is necessary to locate the controller in the remote site, which is connected with the actuator and the sensor through the Internet, as shown in Fig. 4. The Internet has become part of the system in this case. The transmission delay is introduced in both the actuator and sensor communication channels. The author has implemented this structure for the remote design and maintenance of real-time software over the Internet. The detail can be found in the literature [11].

Many existing Internet-based control systems adopt a bilateral control structure, i.e. one controller located in the robot site, another in the operator site, and linked through the Internet as shown in Fig. 5. For example, based on this control structure, robotic tele-operation uses the controller in the robot site to control the slave device, and uses the one in the operator site to control the master device. Usually the controller in the robot site is responsible for the regulation of the normal situations. Once the performance of the controller is degraded due to the disturbance from the environment or the change of the production situation, the controller in the operator site is put in use for tuning the parameters and/or changing the desired input for the controller in the robot site.

4 Design of the Multimedia-Based User Interface

The central design objective for the Internet enabled arm robot control is to enable web clients to directly manipulate the arm robot and appreciate more rapidly what is happening in the physical arm robot located remotely from the users. There are a number of general user-centred interface design principles available [1, 5], like user in control, directness, consistency, feedback and simplicity. This work fully follows these general-principles and focuses on providing a flexible, direct and easy-controlled 3-D interface for the Internet enabled arm robot.

Macromedia Flash has been chosen for the development of the interface. Flash allows users to create full screen animation (flash movie) and interactive graphics. The flash file is saved in a vector-based format, which results in extremely compact files. This feature enables the flash-based animation to be embedded and quickly downloaded in a web page.

The interface for our arm robot is illustrated in Fig. 6, which provides not only individual control buttons to initiate various actions, like starting, stopping, initialising, and logging-out, but also a dynamic simulator for virtually operating the physical arm robot. With this simulator users can control the arm robot off-line, choose the best position for the arm, and then send the automatically generated control command to the physical arm robot by pressing the ‘Send data’ button. The actual position of the arm robot can be viewed from the web camera visual feedback and the graphical feedback is shown in the left hand side in the interface.

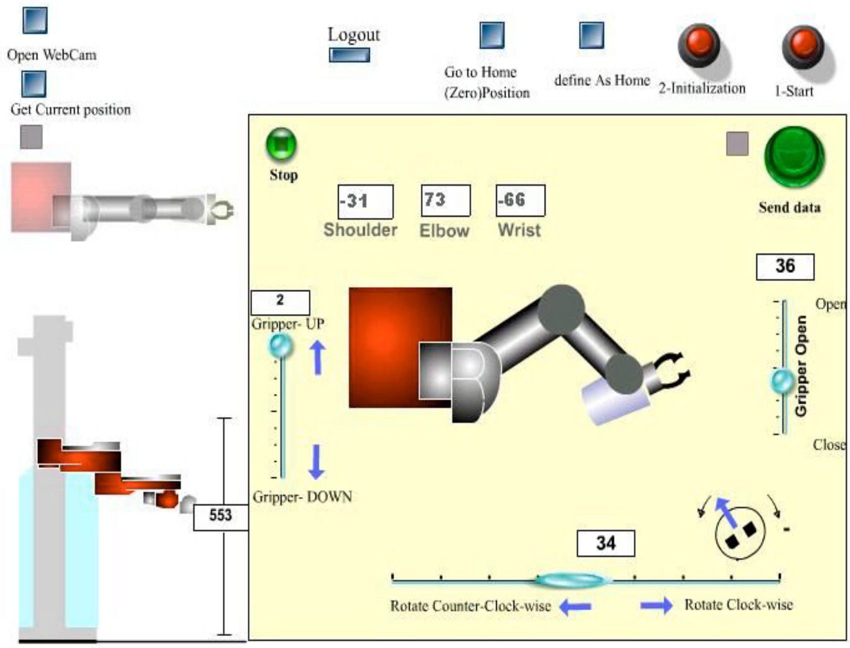


Fig. 6. Web interface for the arm robot designed with Flash

5 Open Loop Control with the Transmission Latency

In this section, the simplest control structure shown in Fig. 3 is chosen for an open loop Internet-based robot control system. The block diagram of the control system is shown in Fig. 7. The reason of adopting this control structure is purely for the simplicity. Total time of performing an operation per cycle is $t_1 + t_2 + t_3 + t_4$, where the four types of time delay are:

t_1 time delay of making control decision by the operator;

t_2 time delay of transmitting the control command from the operator to the robot;

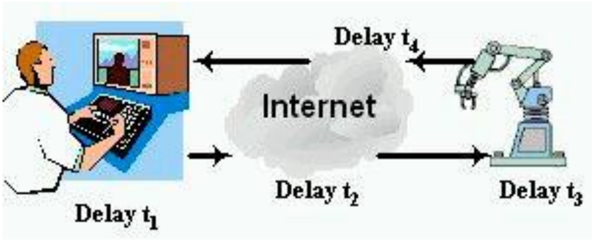


Fig. 7. The block diagram of the Internet-based robot control

t_3 execution time of the robot to perform a primitive action;

t_4 time delay of transmitting the information from the robot to the operator.

If each of the four time delays is a constant, robot control over the Internet has a constant time delay. Unfortunately, t_2 and t_4 are usually unpredictable. Luo and Chen [6] have repeatedly tested the transmitting efficiency of the Internet by sending 64 bytes data continuously from their Web server to different remote servers. The resulting statistics show the Internet not only contains serious and uncertain time delays but also data-loss. In the TCP/IP protocol, once the data is lost the remote site will require a retransmission. This leads to a longer overall delay. The long transmission delay may result in remote control failures in a complex task or, more seriously, endanger the robot and its surroundings.

Under the open-loop control, the operator makes the control decision based on video feedback from the robot. The robot is designed to reject any further control command before the on-going task is completed. Therefore, the time delays t_2 to t_4 only can slow down the execution of the operation cycle, but do not endanger the robot and its surroundings.

6 Implementation

A number of web technologies including HTML, HTTP, PHP, TCP/IP, and Java have been used in the implementation of the Internet enabled arm robot control. HTML is used to build a front-end including a welcome page, a logging-in page, and an interface frame. The interface was established by using Macromedia Flash and embedded in the HTML interface page. The communication between web clients and the web server follows the HTTP communication protocol. A Java program located in the server machine directly controls the arm robot. TCP/IP sockets are used to build communication between the Java program and the web server through a PHP program

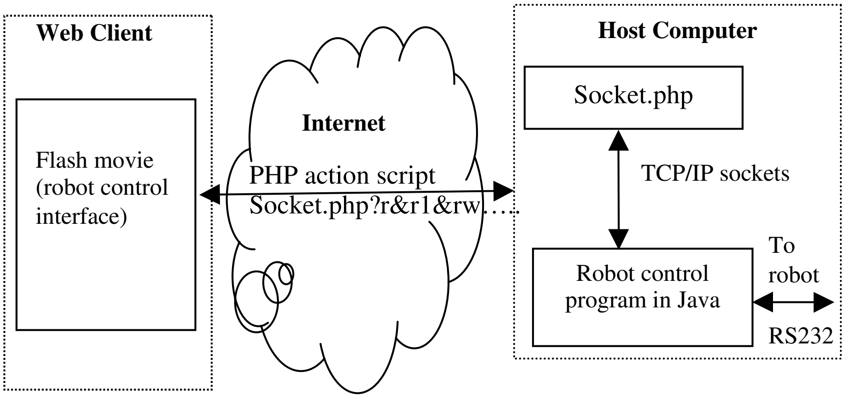


Fig. 8. Information flow of the arm robot control

Socket.php. The PHP program is also used to operate a mySQL database for user management. The flow of information is shown in Fig. 8. The control commands collected from the web client interface are sent to the PHP server by using the following PHP action script:

```
LoadVariableNum (
"http:// pc-robot.lboro.ac.uk/Socket.php?Para1 ="+Para1
+ "&Para2="+Para2 + …);
```

where `'hhtp://pc-robot.ac.uk'` is the web address of the PHP file *Socket.php*, *Para1* and *Para2* are two parameters that are required to transfer to the PHP server and then to Socket.php.

In the Socket.php program the TCP/IP sockets are used to communicate with the Java program, which on-line controls the arm robot. The PHP action scripts for TCP/IP communication include:

```
$connection=fsocketopen(host, port, timeout);
fputs($connection, "move $Shoulder");
…
fclose($connection);
```

The first line is to build the TCP connection to the remote host on the specific port. The '*timeout*' indicates the length of time in seconds to wait before timing out. The second line transfers the control command `'move $Shoulder'` to the host. The third line closes the connection.

User information, including who they are, what they did and when, have been automatically logged in a database for management purposes.

In order to display the visual feedback from the web camera the NetMeeting ActiveX control has been added to the client interface. The visual window is popped out/off by pressing the 'OpenCam' button in the interface shown in Fig. 6.

7 Conclusions

This paper have presented the challenges of design of Internet based control systems and described three general structures for this type of control systems. Various web technologies have been used in the arm robot case study, including HTML, HTTP, PHP, TCP/IP, and Java. Communication is based on PHP pages and TCP/IP sockets. In order to provide a flexible, direct, and easy-use interface for web clients, a multimedia based dynamic simulator of the arm robot is embedded in the interface. Open loop control structure is implemented in the case study. Users control the arm robot offline first through operating the simulator, and then send the automatically generated control commands to the physical arm robot by pressing a confirmation button. Our experimental work shows that operating the arm robot through the simulator at the client site offers a greater opportunity for the users to improve the operability of the physical arm robot over the Internet.

References

1. Alty, J.L.: Multimedia and Process Control Interfaces: Signals or Noise? Transaction of Inst MC, 21 (1999) 181-190
2. Baruch, J.E.F. & Cox, M.J.: Remote control and robots: An Internet solution. IEEE Computing Control Engineering Journal, Feb. (1996) 39-44
3. Copinga, G.J.C., Verhaegen, M.H.G., & Van De Ven, M.J.J.M.: Toward a Web-based study support environment for teaching automatic control. IEEE Control Systems Magazine, August, 20 (2000) 8
4. Han, K. H., Kim, S., Kim, Y. J. and Kim, J. H.: Internet control architecture for Internet-based personal robot. Autonomous robots, 10 (2001) 135-147
5. Hori, S. and Shimizu, Y.: Designing Methods of human interface for supervisory control systems. Control Engineering Practice, 7 (1999) 1413-1419
6. Luo, R.C. and Chen, T.M.: Development of a multibehaviour-based mobile robot for remote supervisory control through the Internet. IEEE transactions on mechatronics, 5 (2000) 376-385
7. Taylor, K. & Dalton, B.: Internet robots: a new robotics niche. IEEE Robotics & Automation Magazine, March (2000) 27-34
8. Yang, S. H. & Alty, J.L.: Development of a distributed simulator for control experiments through the Internet, Future Generation Computer Systems, 18 (2002) 595-611
9. Yang, S. H. & Chen, X.: Dealing with time delay and data loss for Internet-based control systems, IFAC Workshop on Time Delay Systems, Rocquencourt, France, September, CD-ROM (2003)
10. Yang, S. H., Chen, X. & Alty, J.L.: Design issues and implementation of Internet based process control, Control Engineering Practice, 11 (2003) 709-720
11. Yang, S. H., Chen, X., & Yang, L.: Integration of control system design and implementation over the Internet using the Jini technology. Software Practice and Experience, 33 (2003) 1-25
12. Yang, S. H., Tan, L.S. & Chen, X.: Specification and architecture design for Internet-based control systems, Proceedings of the 26th Annual International Computer Software and Applications Conference, IEEE Computer Society, Oxford, UK, (2002) 75-80

Source Estimating Anycast for High Quality of Service of Multimedia Traffic

Won-Hyuck Choi, Tae-Seung Lee[*], and Jung-Sun Kim

School of Electronics, Telecommunication and Computer Engineering, Hankuk Aviation University, 200-1, Hwajeon-dong, Deokyang-gu, Koyang-city, Kyonggi-do, 412-791, Korea
rbooo@korea.com, thestaff@hitel.net, jskim@mail.hangkong.ac.kr

Abstract. Multicast communication techniques can supply the most appropriate infrastructures for multimedia having to carry data of heterogenous types. As a major multicast protocol, the core based tree (CBT) protocol has been concentratively studied. The CBT places a core router at the center of the shared tree and transfers data through the core router. However, the CBT has two problems resulted from centralizing all network traffics into a core router. First, it can raise bottleneck phenomenon at a core router. Second, it is possible to make an additive overhead for processing when a core router is distant from receivers. To cope with the problems, in this paper we seek an intelligent anycast routing protocol. The proposed anycast routing estimates traffic characteristics from multimedia data for each multicast source, and places a proper core router to process incoming traffic based on the traffic information when requests of receivers are raised. This method prevents the additional overhead to distribute traffics because an individual core router uses the information estimated about multicast sources connected to oneself and the statistics for processing traffics are shared with other core routers.

Keywords: internet applications, multicast routing, anycast routing, intelligent traffic distributing

1 Introduction

Recently, the evolving of computer communication has enabled networks to transmit the pervasive multimedia traffics of various types. According to it, efficient methods are required to provide quality of services (QoSs) for video conference, virtual reality, remote medical diagnosis system, video on demand and so on.

The prevailing QoS methods for multimedia services are based on the utilization of transmission line characteristics, compression technologies, and routing algorithms. However, to achieve QoS for real-time multimedia, many problems must be solved in efficiently using resources, easily changing topologies, and additionally expanding network bandwidth. Among the problems, the efficient use of resources has been taken more interest in by reason of economical consideration [1].

[*] The corresponding author will reply to any question and problem from this paper.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 673-678, 2004.

Multicast protocols are a QoS-guarenteed protocol. In multicast, a sender can transmit data to all the participants who have joined in a group by using IP multicast. Evolving out of the existing point-to-point communications, multicast can transmit packets in point-to-many or many-to-point mode. The core based tree (CBT) is a multicast protocol that has gathered intensive attention. The head concept of the CBT is to construct a shared tree that loses little throughput, minimizing the overhead from the locations not balanced in the shared tree. However, the construction of shared trees has the trouble in NP-complete of Steiner minimal tree. This trouble causes to concentrate traffics on a core and place the core on an improper location.

To cope with the problems, we investigate a novel and intelligent anycast protocol to make efficiently use of network bandwidth. Anycast is a variety of multicast to distribute the centralized traffics among plural core routers by using some distributing algorithm. The anycast routing we propose estimates the traffic characteristics from multimedia data for each multicast source, and places a proper core router to process incoming traffic based on the traffic information when receiver requests are raised. This method prevents the additional overhead by distributing traffics, in which an individual core router uses the information estimated about multicast sources connected to oneself and the statistics for distributing traffic is shared with other core routers. To verify the feasibility of the proposed method, simulation and evaluation are conducted using a representative simulation tool.

This paper is organized as follows. In Section 2 we describe briefly the CBT algorithm. In Section 3, the anycast routing protocol adopting an intelligent method is proposed, and the result of simulation for the performance evaluation of the proposed protocol is analyzed in Section 4. Finally, the paper is summarized in Section 5.

2 Core Base Tree

The existing multicast protocols can be divided into the source-based SBT multicast that a tree is constructed for each source and the shared tree (ShT) multicast that multiple sources share a tree [5]. The ShT is denoted as (*, G), where * means the whole sources and G the group. The computational amount of a real tree is O(|G|) independently of the number of sources since the tree is a shared tree. The cost to construct a tree is inexpensive, but a serious traffic delay may be caused by increasing the number of sources. The ShT is suitable for the networks that there are many multicast services and service senders but traffic bandwidth is narrow. The CBT is one of the ShT and has a core router on the center of the shared tree. In contrast to the protocol independent multicast in sparse mode (PIM-SM) protocol that operates through a uni-directional tree and so has constraints on selecting an optimal routing path, the CBT tree operates as a bi-directional tree and has the more flexible extensibility of networks compared with the existing source-based multicast routing protocols.

The CBT constructs a shared tree that minimizes the overhead resulted from the locations not balanced in the shared tree with a little lose of throughput to transmit traffics. However, the trouble in NP-complete of Steiner minimal tree lies in the construction of a shared tree and from that two problems arise to be solved [2], [3].

The first problem is the traffic concentration on a core. This phenomenon is observed when heavy traffics made from many types of video, telnet, ftp and so on are concentrated on a core router at a time. The phenomenon called traffic concentration is depicted in Fig. 1.

The second problem is the poor placement of a core. The ideal location of a core to receive traffic is the center of group. However, if the core is distant from receivers and used independently by them, high bandwidth and large storage are not meaningful anymore and it is even impossible to place an appropriate core. Fig. 2 describes the poor core phenomenon.

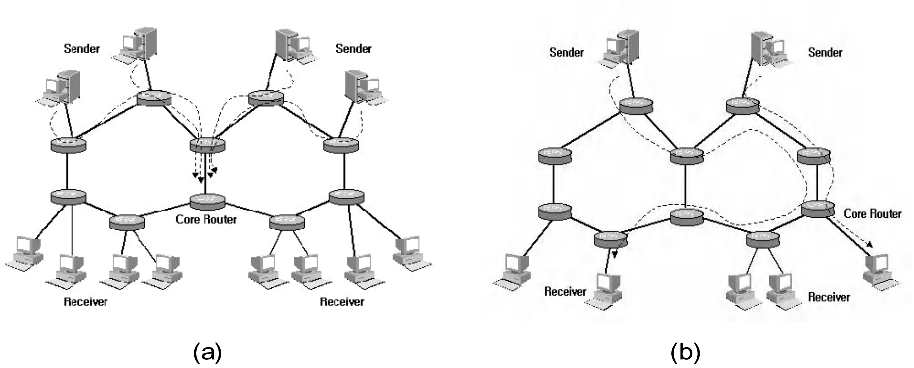


Fig. 1. The two problems to be solved for the CBT: (a) traffic concentration and (b) poor placement of a core

3 Intelligent Anycast Protocol

The intelligent anycast proposed in this paper attempts to solve the two problems raised in Section 2: the traffic concentration and the poor placement of a core. In the anycast the concentrated traffics are able to be controlled, so QoS for real-time transmission of traffics is ascertained by optimizing distribution of cores and the network can be economically managed [6]. The proposed anycast analyzes characteristics of traffics before groups generating the traffics join in cores, and forwards the traffics through the most appropriate core.

The proposed anycast routing goes through two processes. First, entire traffics are grouped into homogeneous traffics according to their characteristics and router tables are generated for all the homogeneous traffics. Second, user profiles are created to manipulate the preference to each source and incoming traffics are estimated from the user profiles. In each anycast router, the router tables generated from the traffic characteristics are accessed through an intelligently and hierarchically established mechanism and routers share user-requested traffics with each other by estimating the traffics. The atomic processes composing the proposed anycast are as follows:

Traffic retrieval: traffic information saved in router tables is retrieved.

User profile: user information is got from the keywords that morpheme analyzer made to catch up user interests generates.

Traffic grouping: router tables are created to group incoming traffics into homogeneous traffics according to their contents. The purpose is to reduce load on network and maximize utilization efficiency of network.

Automatic estimation: the retrieval results from characteristics of source traffics are provided by router tables

4 Simulation and Evaluation

To simulate the intelligent anycast proposed in Section 3, traffic rate on a CBT core is estimated and analyzed when bottleneck situation is made around the CBT core. Based on the theoretical approach, a simulation is conducted to compare the CBT with the proposed anycast and the performances of them are recorded in traffic condition, core links situation, and traffic characteristic.

In the simulation, 11 senders, 6 groups, and up to 4 receivers for each group are provided to record the performance while changing the numbers of senders, groups, and receivers in a group. Each multicast link has bandwidth fixed to 1.5 Mbps and propagation delay to 10 ms. Packet rate flown into routers follows Poisson distribution and the queuing to model arriving intervals of packets and their processing times is with M/M/1 following exponential distribution. The capacity of the queuing according to the size of router buffer is assumed to be infinite. Message forwarding time to process packets in routers is set to 3 ms and idle time of routers to 200 ms. The simulation is conducted on a PC that has 512MB memory and 1.9 GHz Intel Pentium 4 processor, and uses Linux Redhat 7.0 as its operating system. The simulation is implemented with the network simulator version 2 (NS-2) that is wildly used as PC-based simulator.

For the simulation, the topology represented in Fig. 2 is constructed. The proposed anycast routing can catch out the link conditions of cores from the information of router table and protect a core from falling into bottleneck. For performance evaluation of multicast routings, CBT cores of the given topology are compared with anycast cores at bottleneck situation. The units of 210, 512, 1024 and 1280 bytes are selected as the sizes of packets to represent characteristics of multimedia.

The CBT routing and the proposed anycast are evaluated for each size of packets in Fig. 3 and Fig. 4, respectively, as changing the numbers of multicast groups and senders. From the simulation results, when the sizes of packets are 210 and 512 bytes, the queueing delay is a rather superior. But for the sizes of packets, 1024 and 1280 bytes, delay comes to be made just over 20 packets and bottleneck arises at once.

The packet delays for CBT core is displayed in Fig. 3. As seen in the figure, it is noted that after the multicast tree is initialized groups are joined or leaved frequently. As a result, the arrival intervals of packets become short and the traffic is increased. It is possible that this situation give a critical impact on the multimedia QoS for real-time transmission.

In Fig. 4, the delays of the proposed anycast protocol is demonstrated while the CBT routing is retained. Though the anycast is adopted, the increase of the number of packets makes the same delays as those in the Fig. 3 but some difference in generation time of the bottleneck. Nevertheless, the anycast alleviates the queueing delays slowly at the bottleneck when the packets are suddenly concentrated after the multicast tree is joined in and the routing table is updated.

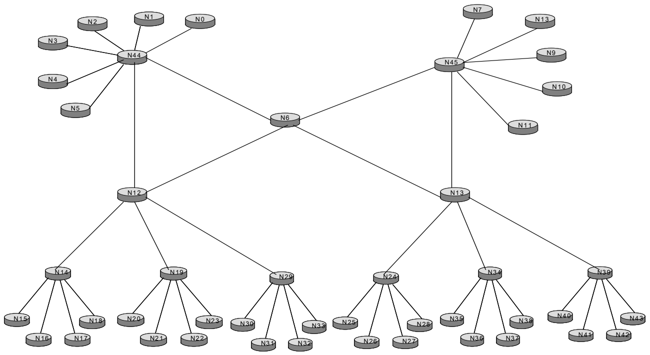


Fig. 2. Topology for the simulation

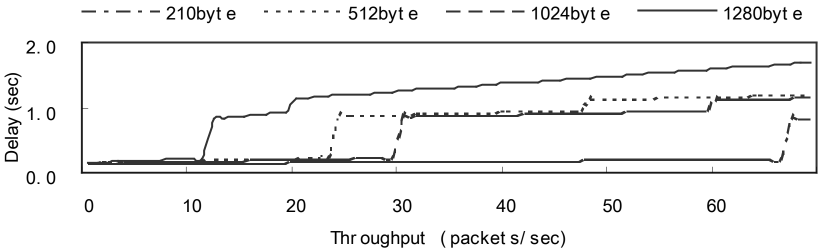


Fig. 3. Packet transmission delays for CBT core

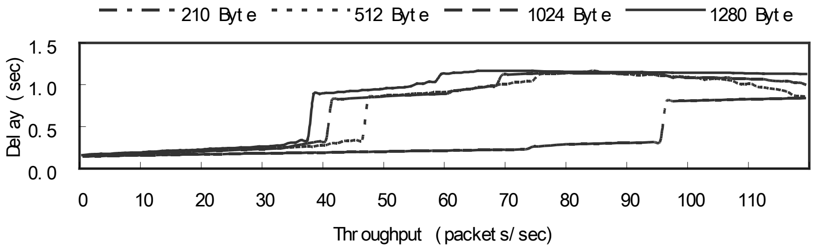


Fig. 4. Packet transmission delays for CBT/anycast core

5 Conclusion

So far with the proposed anycast we have attempted to resolve the two problems, traffic concentration and poor core placement, raised with the CBT multicast protocol by forecasting the characteristics of incoming traffics. As a shared tree routing, the CBT operates effectively on low traffic, but has some trouble on huge traffic. To relieve the problem, we researched the possibility of the intelligent anycast routing to distribute increasing traffics among plural CBT cores. From the results of simulation, it is known that when the proposed anycast algorithm is used for various multimedia services the existing advantages of multicast and QoSs for multimedia traffics can be obtained at the same time.

References

1. Parsa, M, Garcia-Luna-Aceves, J. J.: A Protocol for Scalable Loop-free Multicast Routing. IEEE J. Select. Areas Commun. **15** (1999) 316-331
2. Jia, X. Wang, L.: A Group Multicast Routing Algorithm by Using Multiple Minimum Steiner Trees. Computer Communications (1997) 750-758
3. Ballardie, A.: Core Based Trees (CBT) Multicast Routing Architecture. RFC2201 (1997)
4. Ballardie, A.: Core Based Trees (CBT version 2) Multicast Routing - Protocol Specification. RFC2189 (1997)
5. Moy , J.: Multicast Extensions to OSPF. IETF RFC 1584 (1994)
6. Ettikan, K.: An Analysis of Anycast Architecture and Transport Layer Problems. Asia Pacific Regional Internet Conference on Operational Technologies (2001)

Scheduling Meetings with Distributed Local Consistency Reinforcement

Ahlem Ben Hassine[1], Takayuki Ito[2], and Tu Bao Ho[1]

[1]School of Knowledge Science, Japan Advanced Institute of Science and Technology,
1-1 Asahidai Tatsunokuchi-Machi, Japan
{hassine, bao}@jaist.ac.jp
[2]Graduate School of Engineering, Nagoya Institute of Technology,
Gosiko, Showa-ku Nagoya, 466-8555, Japan
itota@ics.nitech.ac.jp

Abstract. Meeting scheduling (MS) is an important real-world problem. Solving this problem consists in scheduling all the meetings while satisfying all the constraints. However, human nature often has conflicting preferences. The majority of works, dealing with MS problem, allowed the relaxation of the preferences in order to reach an agreement between all the participants, but this is not always possible. To overcome this difficulty, the main contribution of our work consist in trying to satisfy as much as possible users' preferences while taking into consideration their availabilities, and this through a new approach based on the *distributed reinforcement of arc consistency* (DRAC) model. The new approach was implemented and the experimental results show that our approach is scalable and worthwhile to handle especially strong constraints.

1 Introduction

In our daily life, meeting scheduling (MS) is an important problem. It can be defined by the process of scheduling events (meetings), while taking into account several constraints. These constraints are essentially related to the availabilities and preferences of the users who should participate in the meetings. This problem is naturally distributed and cannot be solved by a centralized approach. Solving MS problem involves determining the date, the time and the duration of the meetings[1] that must be held between several users. Each user has his/her available times, timetabling, and preferences. Moreover, each meeting has a priority. Therefore, solving an MS problem means to find a compromise among different human user's requirements.

Many significant research efforts dealing with the MS problem have been proposed in the literature. The initial meeting scheduling researches are based on CSP (constraint satisfaction problem) formalism [10]. The problem is formalized as centralized CSP in which all the users' information is centralized in the same process [1, 3]. These works focused essentially on over-constraint CSPs. However, recent

[1] In the rest of this paper, we use the term date to define the date, time and duration of a meeting and thus to simplify the problem.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 679-688, 2004.

researches adopt an agent-based approach for many reasons. The main reason is that agents can accomplish their tasks through cooperation while allowing the users to keep their privacies. Among these works, [6] focused on using distributed autonomous and independent agents to solve a problem. This work is based on the communication protocol presented in [11].

Sen and al. [13] have proposed another work based on how an application domain for intelligent surrogate agents can be analyzed, understood and represented in order to make these agents able to carry out tasks on behalf of human users, taking into account their environment. Their prior work focused on agents adapting to environmental changes [14], but in [13] their efforts directed towards the integration of user preferences. However, often users preferences are mutually conflicting, so the authors used techniques from voting theory to formally represent and reason with conflicting preferences. Two other methods, using the Partial CSP model [5], were proposed in the literature. The first work proposed by [7] offered a new approach for MS problems using fuzzy constraints. The underlying protocol is called the selfish protocol, where each user tries to maximize its preferences during the negotiation process. The second in [15], used the *distributed valued constraint satisfaction problem* (DVCSP) formalism to model the MS problem.

In all of these works, the basic common point is the relaxation of any users' constraint, even non-availability constraints, in order to get an agreement between all of the agents and consequently solve the problem. However, in real world problems, it is not always permitted to relax the constraints of users. For example, when the user is experiencing relaxing travel, such a constraint would oblige the user to stop his/her travels and to attend the meeting, which is not possible. In addition, none of these approaches tried to maintain any level of consistency during the negotiation process. Reinforcing local consistency techniques allow us to reduce the problem and especially to improve the efficiency of the solving process.

In a different way, our main contribution is to try to more closely reflect real applications while improving the process of scheduling meetings. The proposed protocol is based on *distributed reinforcement for arc consistency* (DRAC) approach [2]. The basic idea is to benefit from the main goal of DRAC in order to reduce the complexity of a meeting-scheduling problem solving process. Thus, the meeting-scheduling problem is viewed as a set of distributed agents in communication, each of them acts on behalf of one user, and maintain in private its personal information. All the agents will cooperate and negotiate by exchanging only relevant information. Thus the final result, i.e. the scheduling of the meetings maintained by agents, is obtained as a consequence of the agents' interactions. In such a manner, all of the agents act in parallel and asynchronously via sending asynchronous messages.

We propose, in this work, to formalize the MS problem as a *valued constraint satisfaction problem* (VCSP) [8] in which two kinds of constraints are considered: hard and soft constraints. The hard constraints (which can never be violated) represent the non-availability of the user, while the soft constraints (which can be violated) represent the preference calendar of a user. Furthermore, each new scheduled event is considered as a hard constraint.

This paper is organized as follows. First we will present the related works. Next, we will present the DRAC model adapted to the MS problem. Then we will present the global dynamic followed by an example of illustration. Finally, we discuss the experimental results and give our conclusions.

2 Formalization of the MS Problem

We propose to formalize a MS problem as a VCSP [8]. Therefore, we first present the CSP formalism followed by VCSP formalism and finally we give our proposed formalization for the MS problems.

A CSP [10] is triplet (X, D, C) composed of a finite set of *n* variables X={X_1, ..., X_n}, each of which is taking values in an associated finite domain D={D_1, ...,D_n} and a set of *e* constraints between these variables C={C_{ij}, ...}; C_{ij} is a constraint between X_i and X_j. The constraints restrict the values the variable can simultaneously take. Solving a CSP consists in finding one or all-complete assignments of values to variables satisfying all the constraints.

This formalism is generalized to the over-constrained problems by giving a weight or a valuation to each constraint reflecting the importance of satisfying it. A VCSP [8] is a quintuple (X, D, C, S, φ) where (X, D, C) is a classical CSP formalism, S=(E, ⊗, ≻) is a valuation structure and φ: C→ E. E is the set of possible valuations; ≻ is a total order on E; ⊥∈E corresponds to the maximal satisfaction, ⊗ is an aggregation operator used to aggregate valuation.

Assume that *A* is an assignment of all the variables of the problem. The valuation of A is defined by:

$$\varphi(A)= \otimes_{c\in C}\varphi(A, c) \text{ where: } \varphi(A, c)= \begin{cases} \bot \text{ If } c \text{ is satisfied by } A \\ \varphi(c) \text{ Otherwise} \end{cases}$$

We have used the VCSP formalization to define the MS problem by (X, D, C, S, φ), in which:

- X={..., X^i_k, ..., X^j_h, ...}, where X^i_k is date requested by the user *i* for the meeting m^i_k.
- D={..., D^i_k, ..., D^j_h, ...}, where D^i_k is the set of possible dates for the meeting m^i_k.
- C is the set of all the constraints of the problem. We divide the set C into two types of constraints: Constraints related to the users and Constraints related to the meetings. For the former type, we can consider:
  - Hard constraints: C_H related to the non-availability of the users,
  - Soft constraints: C_S related to the preferences of the users.

  As for, the second type of constraints, it represents all the *allDiff* constraints [12] existing between each pair of meetings sharing at least the same participant.

  It is noteworthy that for this type all the constraints are considered as hard constraints.
- S is the valuation structure where ⊗ corresponds to +, and ≻ defines the operator more than ">". This operator is used to set a total order among the obtained solutions in the problem.

For each hard constraint $c_H \in C_H$, we associate a weight ⊥, for each soft constraint $c_S \in C_S$ we associate a weight[2] $W^i_{k_l} \in E$, and to each meeting m^i_k we associated a weight[3] $w^i_k \in E$.

[2] It represents the degree of preference of the agent A_i for having the meeting m^i_k at the date d_{k_l}.

[3] This weight defines the priority/importance of m^i_k.

3 DRAC Model for MS

The DRAC model uses two kinds of agents: Constraint agents and Interface agent. Each has its knowledge (static and dynamic knowledge), a local behavior to satisfy, and a mailbox to store incoming messages. The different agents communicate by exchanging asynchronous point-to-point messages. For the transmission between agents, we assume that the messages are received in a finite delivery time and in the same order they are sent.

This model can be "well" adapted to the MS problem. In this problem, each Constraint agent can be considered as a User agent. A User agent must maintain the concerned user's calendars for both availability and preferences.

The acquaintances of an agent consist of all of the agents that must be present in the same meeting, which we call Participant agents (we represent as *participant*(A_i)). Accordingly, in our system an agent is considered a Proposer agent when it has a meeting to schedule. It can be also considered as a Participant agent i.e. it is a participant in another meeting proposed by another agent of the system. Each scheduled meeting that has been registered is considered as a new constraint. Therefore it must be added to the set of constraints maintained by the corresponding agents.

Each agent A_i maintains a VCSP^{A_i} for which the variables $X^{A_i} \in X$ represents the meetings dates to found for its user's set of meetings (we represent as *meetings*(A_i)), while the constraints $C^{A_i} \in C$ represent the non-availability, the preferences of the corresponding user and the *allDiff* constraints involving its meetings. Thus in the proposed model, the constraints $C^{A_i} \in C$ represent the intra-agent constraints for A_i while the inter-agent constraints are represented by equality constraints.

Each attendant has a set of meeting preferences for each particular meeting. The local goal is to schedule meetings such that all its hard constraints C_H are satisfied while trying to maximize the proposer's preferences (*selfish* protocol) using formula1.

$$Max \sum_{k \in \{1..|meetings(A_i)|\}} W^{i}_{k_l} \qquad (1)$$

The global goal is to schedule maximum of the meetings of the users satisfying all the inter-agent constraints. Each agent in the meeting scheduling process tries to satisfy its local goal while maximizing its preferences (selfish protocol).

The Interface agent is an intermediate interface between a user who wants to schedule a meeting and the Constraint agent. It is added in order to create the agents and, most importantly, to inform the users of the result.

4 Global MS Dynamic

The global objective of the proposed approach is to schedule all of the meetings for all of the users while maximizing their local preferences. This dynamic is divided into two steps:

- The first step uses the basic idea of the DRAC approach, which consists in transforming the original MS problem into another *equivalent* MS'. This step is needed to reinforce some level of local consistency [9] (node and arc consistency) in the initial problem.
- The second step is to solve the obtained MS problem while maintaining arc-consistency and this via interactions and negotiations between Participant agents and the Proposer agent. Each Proposer agent tries to find the best solutions for its meetings.

When a user wants to host a meeting, he must run the Interface agent, which will activate the corresponding Proposer agent and make it interact with all of the Participant agents (Fig.1). More than one Proposer agent can be activated at the same time, i.e. in case there are many users that want to schedule their meetings.

Each activated Proposer agent must first reduce the time slots of the corresponding meetings according to its hard constraints (constraints defining the non-availability of the user). This process can be viewed as a local reinforcement of node consistency and aims to reduce the meetings' slot times by eliminating the dates on which the meeting cannot be held (Fig.1 lines 1 and 2), i.e. a meeting cannot be held on a date defined as a non-available date for the user or already planned for another meeting.

```
Begin
1. For each $m^i_k \in meetings(A_i)$ do
2.   Delete from $D_k^{A_i}$ all $d_{kl} \in C_k^{A_i}$;
3.   Delete from $D_k^{A_i}$ all $d_{kl}$ such that $\exists\, m^j_h \in scheduled^{A_i}$ and $X_h^{A_j} = d_{kl}$;
4.   If $D_k^{A_i} = \emptyset$ Then change the meeting calendar $D_k^{A_i}$ of $m^i_k$;
5.             Else For each $A_j \in participant(m^i_k)$ do
6.                    Send($A_j$, self, "ReduceCalendar:$D_k^{A_i}$ for: $m^i_k$");
End
```

Fig. 1. Process executed by each Proposer agent Ai

If the slot times of a meeting become empty after reduction, i.e. the corresponding user is not available for all of the proposed dates for this meeting. The slot times of the later meeting must be changed (Fig.1 line 4). Otherwise, the Proposer agent must send the obtained reduced slot times for all of the meetings to schedule to all of the Participant agents (Fig.1 line 5).

Each Participant agent, that has received this message, starts first by eliminating both, the non-viable dates from the received slot times of the meetings, i.e. dates that correspond to its non-availability, and all the dates taken by the already scheduled meetings (Fig.2 lines 1 and 2). Then returns the obtained slot times to the sender agent (Fig.2 line 3). The Proposer agent collects first all the received reduced slot times (Fig.2 line 4). Then, starts by scheduling its meetings. It tries to first finds the proposal that maximizes its preferences (Fig.2 lines 6 and 8) and then sends it to the concerned acquaintances. In the case where the Proposer agent cannot find a solution to this problem, it will change the slot time of this meeting (Fig.2 line 7).

Each agent, that has received this proposal, must first check if it has, meanwhile, accepted another proposal for the same date. In the negative case, the agent will first update its hard constraints by adding the new proposal (Fig.2 line 14), then update the dates of its not-yet-scheduled-meetings by eliminating the dates that correspond to the same date of the just scheduled meeting, i.e. in order to maintain the arc-consistency. Finally inform the Proposer agent of his agreement (Fig.2 line 15).

However, if the agent has another meeting already scheduled at the same time as the proposed meeting, it must send a negative answer to the Proposer agent and ask it to change its proposal (Fig.2 line 13). Accordingly, each agent that has proposed a meeting and received at least one negative answer must change its proposal (Fig.2 line 18). Consequently, this agent must decrease its degree of preferences (Fig.2 line 19) and the same process is repeated until an agreement is reached among all of the participants or, after testing all of the solutions and no agreement is reached. In the later, case the Proposer agent must inform the participants that the meeting is cancelled.

```
ReduceCalendar:D for:m
1.  Delete from D all $d \in C_k^{A_i}$;
2.  Delete from $D_k$ all $d$ such that $\exists\ m^j_h \in scheduled^{A_i}$ and $X_h^{A_j} = d$;
3.  Send(Sender, self, "Reply:D for:m");
End

Reply:D for:m
4.  SetD ← setD ∪ D;
5.  If Size(SetD)=|participant($m^i_k$)|
6.  Then D ← $\bigcap_{i \in \{1..|SetD|\}} SetD[i]$;
7.        If D=∅
8.        Then Change the meeting calendar and restart;
9.        Else Choose from D the date d that verifies formula 1.
10.            For each $A_j \in$ participant($m^i_k$) do
11.              Send($A_j$, self, "ReceiveProposal:d for:$m^i_k$");
End

ReceiveProposal:d for:m
12. ok ← true;
13. If ($\exists\ m^j_h \in scheduled^{A_i}$ and $X_h^{A_j} = d$) Then ok ← false;
14. If (ok=true) Then Add($scheduled^{A_i}$, ($m^j_h$, d));
                        Delete d for each non-scheduled meeting;
15. Send(Sender, self, "Response:ok for:m");
End

Response:ok for:m
16. setRep ←setRep ∪ ok;
17. If Size(SetRep)=|participant($m^i_k$)|
18. Then If ∃SetRep[i],i∈{1.. |SetRep|}such that SetRep[i]=false;
19.       Then Choose another date d';
20.            For each $A_j \in$ participant($m^i_k$) do
21.               Send($A_j$, self, "ReceiveProposal:d' for:$m^i_k$");
22.       Else For each $A_j \in$ participant($m^i_k$) do
23.               Send($A_j$, self, "Confirmation:d for:$m^i_k$");
End
```

Fig. 2. Main procedures executed by each agent A_i

The aforementioned dynamic resumes until the system reached its stable equilibrium state. This state can be defined as the satisfaction of all agents in the system. An agent is satisfied when it has scheduled all of its meetings or proved that some of them could not be held. The detection of the stable equilibrium state is achieved by using the well-known algorithm of [4], a state, where all the agents are waiting for a message and there is no message in the transmission channels.

We should emphasize on the fact that in this paper we assume on the one hand that each newly scheduled meeting will be considered as a hard constraint, on the other hand, each agent perform a selfish protocol. This choice is used in order to avoid dynamic changes and especially to escape from infinite processing loop. This work can be considered as the first version of the proposed approach. Our future work includes the integration of the dynamic process and the expected protocol will be focused on maximizing the utility of all the agents of the system.

5 Experimental Comparative Evaluations

To evaluate the proposed approach, we have developed the multi-agent dynamic with actalk, an object oriented concurrent programming language using the Smalltalk-80 environment. In our experiment, we generated random meeting scheduling problems. The parameters used for a meeting problem are: number of agents in the system n, number of meeting per agent m, number of participants in a meeting p, number of hard constraints per agent c_H, number of initial soft constraints per agent c_S, maximal duration of an event d, weights for the soft constraints $W^i_{k_l}$, and weights of the meetings w^i_k (the weight of each hard constraint is equal to 1).

In order to compare our approach with that of [15], we used the same parameters to run both algorithms on randomly generated examples. We must note that the approach in [15] presents some restrictions towards on the one hand; the handle of the hard constraints i.e. all the constraints could be relaxed by this approach and on other hand the discrimination between meetings. This approach processes all the proposed meetings with the same importance neither independently of the proposer nor of the attendants. However in the real world, the meetings are not equivalent. For this reason we have brought to our consideration the notion of priority for meetings in our formalization by associating a weight w^i_k to reflect its greatness. Our approach tries then, in its solving process, to schedule *at first* the most important meeting maintained by each agent (unlike the approach in [15]). In that manner, we attempted to describe ideally the real world meeting scheduling problems. Therefore two kinds of experimentations are given in this section.

For the first kind, we assume that for each generated problem, we have only soft constraints. We carried out the two approaches on the same meeting instances with: n=10, $m\in\{3, 4, 5\}$, p = 7, $c_S\in\{20, 40, 60\}$, d= 60, and $W^i_{k_l}\in[0..1]$, $w^i_k\in[0..1]$ were randomly chosen. The initial calendar in each problem is equal to 60.

Table 1 shows the obtained mean results for the ratio of CPU time of the approach in [15] divided by the CPU time of ours. In order to analyze these results, let us consider the two cases <3; 20> and <4; 20>. At first glance, it seems that the approach in [15] is better than our approach, i.e. it requires less CPU time in these two cases. But this can be justified by the fact that for this kind of problem there are not many constraints in the problem, so the approach in [15] can rapidly find a solution for each meeting without relaxing constraints and thus without causing iteration on the same meeting. However in our approach we try to find the solution that maximizes the user's preferences (not the first solution). Therefore we must check all the possible dates for each meeting and also in our approach, we try to maximize each user's preference for the most important meetings. In all the other cases, the approach in [15] takes more time than our approach because both the number of constraints and the number of meetings grow. Furthermore, in our approach each agent tries to perform all its meetings in parallel while for [15] approach, it is done in a sequential manner.

Table 1. Ratio of mean results of the CPU time for meeting problem without hard constraints (10 instances are generated for each <*m*; c_S>)

	<3; 20>	<4; 20>	<5; 20>	<3; 40>	<4; 40>	<5; 40>	<3; 60>	<4; 60>	<5; 60>
Ratio CPU	0.75	0.73	1.01	1.60	2.46	4.15	3.99	5.88	7.77

Thus, when the number of meeting constraints grows, the probability increases, of getting the same dates for the meetings. Thus, the number of relaxed constraints by the approach in [15] increases leading to more iterations for the same meeting and then to an increase in the CPU time.

As for the second kind of experimentations i.e. to appraise the greatness of the reinforcement of local consistency in the solving process, we have choose to measure the percentage of reduction made by the first step of our approach. For this purpose, examples including hard constraints, were randomly generated with n=10, m=3, $p\in\{7, 5, 3\}$, $c_H\in\{20, 30, 40, 50\}$. For each pair <p, c_H> we generated first 10 instances. Then, we ran each instance 10 times and we measured the average of the achieved results. These results are expressed in term of three criteria: the CPU time spent by each of the both approaches, the percentage of scheduled meetings and the percentage of reduced soft constraints performed by the first step of the proposed approach. To this end, we have introduced some modifications to the approach in [15] to make it worthwhile for both hard and soft constraints.

Table 2. Mean results obtained by the approach in [15] for meeting problems with hard constraints (10 instances are generated for each <*p*; c_H>)

	<7; 20>	**<7; 30>**	**<7; 40>**	**<7; 50>**	**<5; 20>**	**<5; 30>**
CPU Time	1932.56	1508.82	1144.96	875.39	1352.21	1133.22
% Meetings	30.00	10.00	3.33	0.00	63.33	26.67
	<5; 40>	**<5; 50>**	**<3; 20>**	**<3; 30>**	**<3; 40>**	**<3; 50>**
CPU Time	822.16	620.36	638.82	578.85	538.08	486.96
% Meetings	3.33	0.00	100.00	80.00	30.00	3.33

We carried out the two approaches on the same meeting examples. Table 2 and Table 3 show the achieved mean results, of respectively the approach in [15] and our approach, in term of the CPU time and percentage of scheduled meetings. In addition, Table 3 gave also the percentage of the reduced soft constraints accomplished by our approach. These results show that our approach requires less CPU time than approach [15]. This is can be elucidated by the fact that the first step is useful in order to discard the dates that cannot be in any solution and consequently to avoid exploiting them in the solving process leading to CPU time consumption. For example in the case of 7 participants and 50 hard constraints, the problem is over-constrained and thus no meetings can be planned, i.e. no agreement can be reached between all the attendants. So our approach can discover merely the absence of solution from the first step, and before starting the solving process.

Table 3. Mean results obtained by our appoach for meeting problems with hard constraints (10 instances are generated for each <*p*; c_H>)

	<7; 20>	**<7; 30>**	**<7; 40>**	**<7; 50>**	**<5; 20>**	**<5; 30>**
CPU Time	383.9	352.1	324.9	298.1	394.7	348.3
% Meetings	13.33	3.33	3.33	0.00	47.33	10.00
% c_H Reduction	96.42	99.22	99.83	100	91.75	98.67
	<5; 40>	**<5; 50>**	**<3; 20>**	**<3; 30>**	**<3; 40>**	**<3; 50>**
CPU Time	329.1	305.4	308.4	307.8	316.3	293.4
% Meetings	3.33	0.00	96.67	50.00	10.00	3.33
% c_H Reduction	99.83	100	80.83	94.5	98.5	99.67

Nevertheless, for the percentage of the meeting scheduled, the approach in [15] planned, for some cases, more meetings than our approach. This is vindicated by the fact that for our approach we tried to plan at first the most important meeting. For example, in the case <7; 20> the 30% of the meetings scheduled by the approach in [15] may contain the most important meetings in the problem or may not. But for our approach we are sure that the 13.33% of the meetings scheduled are the most important because they are chosen at first to be processed using their weights.

We can conclude that our approach is a scalable approach that outperforms the approach [15] and this especially when the number of meetings increases. We must note also that our approach seems to be more appropriate to the real-world application by dealing especially with strong constraints (i.e. inequality) and by bringing forward consideration the discrimination between the proposed meetings. In addition, the first step of the proposed approach can fill a premature detection of the impossibility for reaching any agreement between all the participants and this by maintaining arc-consistency.

6 Conclusion

The objective of this paper is to propose a new approach for meting scheduling (MS) problems that reflects real-world applications. Therefore, we have considered, in our model, two kinds of constraints to model the users' requirements: hard constraints to model the non-availability of a user and soft constraints to define his preferences.

The underlying multi-agent architecture associates a User agent to each user and makes them interact by sending point-to point messages containing only relevant information to keep their privacy. The basic idea of this approach consists of two steps. The first reduces the initial problem by reinforcing some level of local consistency (node and arc consistency). The second step solves the resulting meeting scheduling problem.

This approach was implemented with actalk under the Smalltalk-80 environment and compared with an existing approach in literature, [15], in mean CPU time and percentage of scheduled meetings by using randomly generated problems. The obtained results show that our approach is scalable and worthwhile to process strong constraints. In addition, in order to show the importance of the first step, i.e. reduction step, we have made other experimentation to measure the percentage of the non-viable values discarded from the meetings' calendars. The obtained results proved that this process is well appropriate for reducing MS problem and consequently the search space without loss of solutions.

In future work, we will try to improve this approach by integrating the propagation process to be able to look toward dynamic changes. Then, we will try to implement the improved approach in a multi-processor platform.

References

1. Abdennadher, S., Schlenker, H. Nurse scheduling using constraint logic programming. In Proc. of IAAI-99, pp 838-843, 1999
2. BenHassine, A., Ghedira, K. How to Establish Arc-Consistency by Reactive Agents. In Proceedings of ECAI'02, 2002.
3. Brakker, R.R., Dikker, F., Tempelman, F., Wognum P. M. Diagnosing and solving over-determined constraint satisfaction problems. in Proc. of IJCAI-93, pp. 276-281, 1993.
4. Chandy, K.M., Lamport, L. Distributed snapshots: Determining global states of distributed systems. TOCS, 3(1): 63-75, Feb.1985.
5. Freuder, E.C., Wallace, R.J. Partial Constraint Satisfaction. Artificial Intelligence, 58 (1-3),21-70, 1990.
6. Garrido, L., Sycara, K. Multi-Agent Meeting Scheduling: Preliminary Experimental Results. In Proceedings of ICMAS'96, 1996.
7. Luo, X., Leung, H.F., Lee, J.H-M. Theory and Properties of Selfish Protocol for Multi-Agent Meeting Scheduling Using Fuzzy Constraints. In Proceedings of ECAI'00, 2000.
8. Lemaitre, M., Verfaillie, G. An Incomplete Method for Solving Distributed Valued Constraint Satisfaction Problems. In Proceedings of AAAI-97 Workshop on Constraints and Agents, 1997.
9. Mackworth, A.K. Consistency in network of relations. Artificial Intelligence, 8 99-118, 1970.
10. Montanari, U. Networks of constraints: fundamental properties and applications to picture processing. Information Scisnces, 7, p:95-132, 1974.
11. Sycara, k., Liu, J.S. Distributed meeting scheduling. In Sixteenth Annual Conference of Cognitive Society, 1994.
12. Stergiou, k., Walsh, T. The Difference All-Difference Makes. In the Proceedings of IJCAI'99, p 414-419, 1999.
13. Sen, S., Haynes, T., Arora, N. Satisfying User Preferences While Negotiating Meetings. International Journal of Human-Computer Studies, 47, 407-427, 1997.
14. Sen, S., Durfee, E.H. On the design of an adaptive meeting scheduler. In Proc. of the Tenth IEEE Conference on AI Applications, pages 40-46, 1994.
15. Tsuruta, T., Shintani, T. Scheduling Meetings using Distributed Valued Constraint Satisfaction Algorithm. In Proceedings of ECAI'00, 2000.

Potential Causality in Mixed Initiative Planning[1]

Yousri El Fattah

Rockwell Scientific
1049 Camino Dos Rios
Thousand Oaks, CA 91360
yelfattah@rwsc.com

Abstract. The paper presents an approach for reasoning about potential causality in plans authored directly by humans in a mixed initiative framework. The approach uses only the temporal ordering of the actions and the task structure of the plan. The term potential is used to emphasize the uncertainty in the causal ordering since no requirement is made on the existence of a complete domain theory as in standard partial order planning. The core contribution of the paper is a formalization and algorithm for extracting a parsimonious description of a potential causality relation, which is presented to the modeler as a representation of the candidate space of sets of causal links consistent with the authored plan. The paper also discusses an implemented system based on this algorithm, and its application in the context of execution.

1 Introduction

Mixed initiative planning (MIP) is a collaborative planning paradigm in which humans and machines collaborate to build effective plans more quickly and with greater reliability [2]. Human modelers and planning algorithms need to interact more effectively in order to develop safe efficient plans, and tools are required that ease the task for the modeler. This paper proposes an approach for supporting ease of plan entry during MIP in which a human planner authors a plan by formulating the planning tasks and selecting and ordering the actions. The approach takes the plan authoring input and generates a concise description of a set of causal links for the plan. This causally linked plan can then be used for execution monitoring and plan revision management.

There has been work on capturing plan rationale and causal structure recovery of plans in the context of partial order planning [3], case-based planning [4], and MIP [5]. A plan rationale aims to explicate why a plan is the way it is or the reason as to why the plan decisions are taken. Unlike previous work which requires complete domain theory of the planning domain, our approach can generate potential causal links

[1] The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory under contract No. F30602-00-C-0038. The views and conclusions contained herein are those of the author and should not be interpreted as representing the official policy or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 689-698, 2004.

in the plan by analyzing the temporal ordering and the task structure of the plan. Requiring complete domain models or expecting the human planner to specify every condition required by the plan operators is not realistic in large real-life planning applications such as in military and space. Our approach complements another direction of research in AI planning focused on qualitative reasoning about plans [6]. In that research the focus is on qualitative cause-effect relations and on calculus for reasoning about change. Our work is complementary in the sense that it offers an approach for initial causal models that can be supplemented with qualitative knowledge to address the problem of lack of complete domain models.

In automated planning there has been work on the problem of removing unnecessary orderings in a total order plan (linear plan) in order to produce a □least constrained□ [7] or □shortest parallel execution□ [8] partial-order plan. B□ckstrm [9] shows that the problem is generally NP-hard. Our work differs in that our partial order is representing potential causal links without requiring a domain theory and an automated plan validity test as in [9].

The core contribution of the paper is a formalization and algorithm for extracting a parsimonious description of a potential causality relation, which is presented to the modeler as a representation of the candidate space of sets of causal links consistent with the authored plan. The paper also discusses an implemented system based on this algorithm, and its application in the context of execution.

2 Problem Statement

Plan authoring in our framework is a triple (T, P, ρ). *T* is a rooted directed tree whose nodes *V(T)* are the plan tasks. The root of the tree is the top-level task and the parent-children relation represents task decomposition. *P* is a totally ordered (t.o.) plan $P = \langle a_1, \ldots, a_n \rangle$, represented by the pair $\langle A, \prec_P \rangle$ where A is the set of actions and $\prec_P$ is a total order defined by the t.o. plan: for $1 \le i, j \le n$, $a_i \prec_P a_j$ iff $i < j$. $\rho \subset A \times V(T)$ is a binary relation that associates the actions with the tasks.

Given a plan authoring input (T, P, ρ) the problem we pose is to infer a partial-order plan (p.o. plan) given by a tuple $\langle A, \prec \rangle$ where $\prec$ is a partial order on A[2]. The partial order represents a parsimonious description of a potential causality relation on the actions consistent with the t.o. plan *P*. We represent the partial order $\prec$ by a directed acyclic graph (DAG), G=(A,E), called causal graph, such that there is a directed path from a_i to a_j in G iff $a_i \prec a_j$. The directed edges $(a_i, a_j) \in E$ represent both □causal links□and □threat resolution links□as commonly defined in partial-order

[2] A partial order is a binary relation that is irreflexive, asymmetric and transitive. A binary relation on a set X is a subset $R \subset X \times X$. The relationship is irreflexive if for all (x,x) in $X \times X$, then $(x,x) \notin R$. The relationship is asymmetric if for all $(x,y) \in R$, then $(y,x) \notin R$. The relationship is transitive if for all x, y, z, if $(x,y) \in R$ and $(y,z) \in R$ then . $(a_i, a_j) \in E$ $(x,z) \in R$

planning [1]. A causal link from a_i to a_j means a_i achieves some precondition for a_j while a threat resolution link is a temporal ordering that must be enforced to protect conditions achieved by some causal link. The causal graph *G* represents the p.o. plan which is a compact representation for a set of t.o. plans that can be obtained as the topological sort of *G*. A topological sort of *G* is a sequence obtained by ordering the nodes *A* such that if $(a_i, a_j) \in E$ then a_i is before a_j in the sequence.

Definition 1 Given a plan authoring input $\Pi = (T, P, \rho)$; $P = (A, \prec_P)$ we say a partial order $\prec$ on *A is a* Π-compatible *potential causality* order if it satisfies the following postulates: (i) the total order $\prec_P$ obeys the partial order $\prec$,i.e., $\prec \subseteq \prec_P$ (ii) if two actions a_i, a_j belong to same task $t \in V(T)$, i.e., $(a_i, t) \in \rho$ and $(a_j, t) \in \rho$ then the actions must be ordered following the t.o. plan, i.e., $a_i \prec a_j$ if $i < j$ and $a_j \prec a_i$ if $j > i$ (iii) if two actions a_i, a_j belong to tasks $t_i, t_j \in V(T)$,i.e., $(a_i, t_i) \in \rho$ and $(a_j, t_j) \in \rho$ such that either t_i or t_j is an ancestor of the other in T, then the actions must be ordered following the t.o. plan, i.e., $a_i \prec a_j$ if $i < j$ else $a_j \prec a_i$.

The intuition behind the potential causality postulates is as follows. The temporal ordering of the actions for each task guarantees that each action has effects that are required by the preconditions of the later actions in the same task and therefore provide sufficient condition on the causal ordering. The tasks whose actions do not overlap (do not share actions) are likely to be independent and could be executed in any order. Tasks that are ancestor (or descendant) of one another in the tree are potentially inter-dependent and therefore their temporal ordering should be preserved by the causal ordering. Potential causality provides a space of partial orders from the most to least constrained. The most constrained order is the input plan ordering $\prec_P$ which is a complete ordering. The least constrained causal order can be defined using optimization criteria such as set minimality as defined next.

Definition 2 Given two Π-compatible potential causality orders $P = (A, \prec)$ and $P' = (A, \prec')$ we say that P' is less constrained than *P* iff $\prec' \subset \prec$. A Π-compatible potential causality order $P = (A, \prec)$ is minimal iff there is no other Π-compatible potential causality order $P' = (A, \prec')$ such that P' is less constrained than *P*.

Example 1. Consider the planning problem of putting shoes and socks. A plan authoring input will have: (i) a tree T with a root node and two leaves which are the primitive tasks right foot and left foot; (ii) a t.o. plan e.g., <start, left sock, left shoe ,right sock, right shoe, finish>; (iii) mapping of actions to primitive tasks: {(start ,plan),(finish, plan),(left sock, left foot) ,(left shoe, left foot), (right sock, right foot), (right shoe, right foot)}. Here start and finish are dummy actions to encode the initial state and the goal state for the plan task which is the root node of the tree. It is easy to see that a minimal potential causality order is represented by the causal graph having the directed edges {start→right sock, start→left sock, right sock →right shoe, left sock →left shoe, left shoe →finish, right shoe →finish}. The graph is a compact representation of 6 possible t.o. plans.

We now describe a polynomial time algorithm that computes a minimal potential causality graph (PCG) from a plan authoring input. The algorithm has two main steps. In the first step it computes a minimal potential causality order $\prec$ starting from the complete input plan order $\prec_P$, and in the second returns a directed acyclic graph (DAG) from the transitive reduction of the partial order. The transitive reduction can be computed in $O(|A| \cdot |\prec|)$.

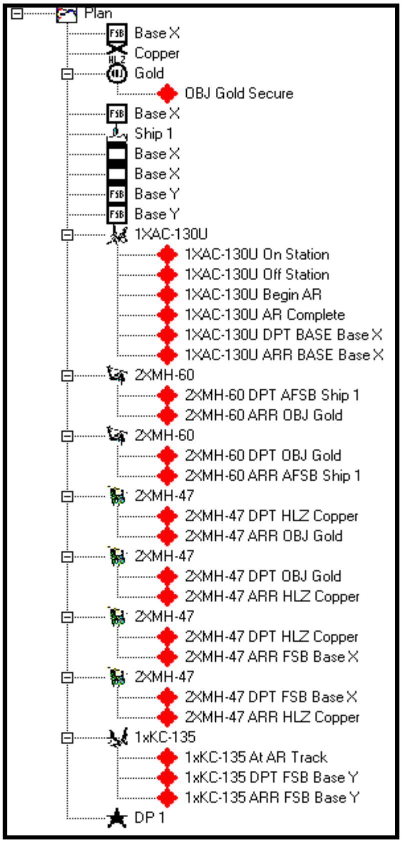
```
Algorithm PCG
Input: authored plan $\Pi=(T,P,\rho)$; $P=(A,\prec_P)$
  begin
  $\prec \leftarrow \prec_P$ //initialize partial order
  //compute minimal portential causality order
  for each $(a,a') \in \prec$
    $S=\{t \in V(T) \mid (a,t) \in \rho\}$//tasks for a
    $S'=\{t' \in V(T) \mid (a',t') \in \rho\}$//tasks for a'
    violation←false
    repeat while not violation
      for each $t \in S$ and $t' \in S'$
        if $S \cap S' \neq \emptyset$ then violation←true
        else if $t' \notin desc(t) \wedge t \notin desc(t')$
             then violation ←true
     loop
     if ¬ violation then $\prec \leftarrow \prec - \{(a,a')\}$
  //causal graph as transitive reduction
  return minimal DAG G=A,E) such that there is a
    directed path from a to a' iff $a \prec a'$
end.
```

3 Application

The approach presented here is applied to a real-life domain in Special Operations Forces (SOF) planning. The ontology of the domain is based on the notion of movement of assets from one place to another. Places represent fixed geographic locations over spans of time and the assets represent resources such as aircrafts, ships, helicopters, vehicles etc. The domain is represented by a set of templates that include descriptions of places, movements, and events.

A plan is composed of any number of events, movements, and places. Figure 1 shows an example of a task tree with nodes representing tasks and the leaves representing actions. Icons are used to identify the assets used in a movement, e.g., AC-130U, and the role of a place, e.g., Base or Ship. A movement has a departure and an arrival events marking its start and end. The start of a movement references an origin place and the end a destination place.A place or a movement can have any number of events bound to it. An event has attributes that include time to specify when the event should occur, and a call time to specify when the event actually occurs during execution.

Plan authoring in this domain consists of a human planner graphically constructing a plan time diagram, like the one shown in Figure 2. The diagram is created with domain specific temporal plan editor having icons representing the various templates in the domain. The plan can be summarized as two black-hawk (MH60s) flying to objective from ship to perform a "secure operation" before the arrival of a Chinook (MH-47). The Chinook flies to objective from a forward staging base (FSB) via helicopter landing zone (HLZ) then returns to FSB. A gunship (AC-130U) is to perform the fire support to protect the Chinook flight from FSB and the black-hawks arrival at objective. The fire support gunship requires aerial refueling (AR) by a KC-135.



```
Plan
    Base X
    Copper
    Gold
        OBJ Gold Secure
    Base X
    Ship 1
    Base X
    Base X
    Base Y
    Base Y
    1XAC-130U
        1XAC-130U On Station
        1XAC-130U Off Station
        1XAC-130U Begin AR
        1XAC-130U AR Complete
        1XAC-130U DPT BASE Base X
        1XAC-130U ARR BASE Base X
    2XMH-60
        2XMH-60 DPT AFSB Ship 1
        2XMH-60 ARR OBJ Gold
    2XMH-60
        2XMH-60 DPT OBJ Gold
        2XMH-60 ARR AFSB Ship 1
    2XMH-47
        2XMH-47 DPT HLZ Copper
        2XMH-47 ARR OBJ Gold
    2XMH-47
        2XMH-47 DPT OBJ Gold
        2XMH-47 ARR HLZ Copper
    2XMH-47
        2XMH-47 DPT HLZ Copper
        2XMH-47 ARR FSB Base X
    2XMH-47
        2XMH-47 DPT FSB Base X
        2XMH-47 ARR HLZ Copper
    1xKC-135
        1xKC-135 At AR Track
        1xKC-135 DPT FSB Base Y
        1xKC-135 ARR FSB Base Y
    DP 1
```

Fig. 1. Task tree

We built a tool that implements our approach for inferring potential causal links in the SOF planning domain. The tool, called Causal Modeler (or CModeler), computes partial orders between the plan events incrementally while the user is plotting the temporal plan diagram. CModeler maintains two data structures: (a) tree depicting the template hierarchy, and (b) directed graph representing a view of the causal relationships. At each step in plan authoring a user can incrementally modify a plan by adding a new template, deleting or editing an existing template. CModeler processes the user actions and recomputes the tree and graph data structures based on the potential causality inference. Having an automated tool for default causal modeling has filled a need in this SOF planning domain for eliciting and capturing causal knowledge during the plan authoring. The causal views computed by CModeler are linked to the temporal and spatial plan views so that changes in one view can propagate to the other views. This is done by representing and capturing the temporal plan constraints and the causal dependencies between the actions in the plan. By viewing the default causal graph a user is able to determine if causal links are missing or existing ones are spurious and can amend the default model by appropriately adding or deleting causal links.

Our approach produces □default□ causal models with incomplete information and the model representation improve as additional constraint specification and knowledge sources about causal links become available. For example the default causal model shown in Figure 3 (the arcs with thin lines) has some of the causal relationships unspecified. Those are the relationships between the fire support and their objectives and between the aerial refueling and the fire support. They are missing in the default model because no constraints were specified between their templates. To amend the default causal graph the user will add those causal links as shown in Figure 3 in thick lines. The Figure shows 6 causal links added to the default model. Link 1 says the start of the fire support necessitates the start of aerial refueling movement. Link 2 says that being at aerial-refueling track enables the refueling action. Links 3, 4 specify the causal condition for the supported missions to begin. The causal links say the fire support □on-station□ event enables the two Blackhawk to depart Ship and the Chinooks to depart drop zone Copper. Links 5, 6 specify the causal condition when the fire support is to end. The links say that the departure of both the Chinooks and the Blackhawk from objective Gold enables fire support □off-station□ event.

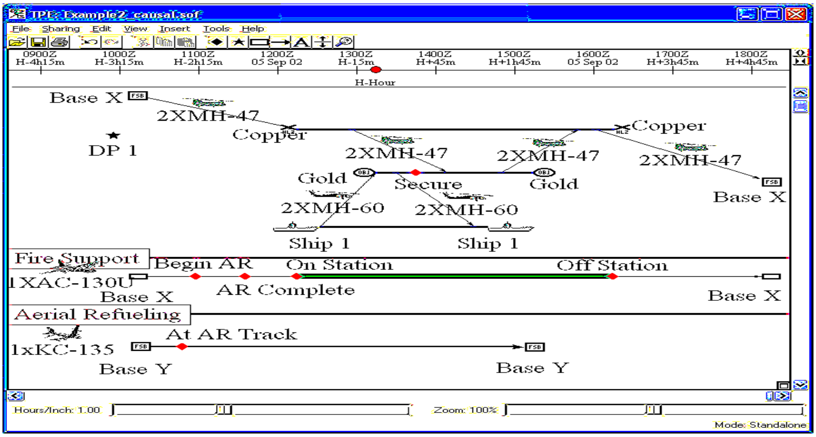


Fig. 2. Plan authored by domain expert in temporal view.

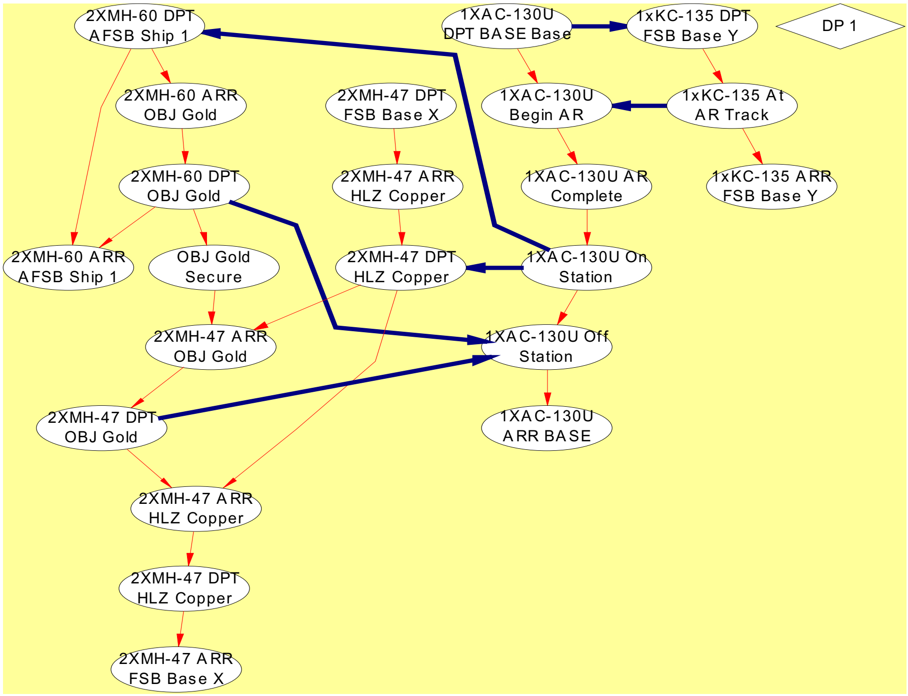


Fig. 3. Potential causality graph with thick arcs added by the modeler.

A principal use of the causal model in the SOF planning application is to perform plan reconfiguration for supporting dynamic plan execution. Most plans do not execute according to schedule and when this occurs there is limited time to respond. When changes occur during execution the task of plan reconfiguration consists in determining all actions impacted by the change and re-computing the plan schedule to re-satisfy the constraints on the affected actions. The affected actions can be determined by performing a reachability analysis on the causal graph. For example, if an action such as the departure of an aircraft is delayed then we must delay all its causally-dependent actions (its descendants in the causal graph), e.g., the arrival of the aircraft at destination and the arrival-dependent chain of events. Note that those descendants are not necessarily contiguous in the plan execution checklist making the determination of causal impact a complex process especially for plans with large number of events.

```
Algorithm PR
Input: checklist L; causal graph G=(A,E).
Output: reconfigured checklist L'.
  begin
  S←∅  //execution state
  <$a_1$,…,$a_n$>←topological sort of G
  for i=1 to n do
   if (calltime($a_i$,L)=?) //not yet executed
```

```
    then if //has some parent in abort state
        (x in pa($a_i$,G): state(x,abort) in S)
      then //set action state to abort
        S$\leftarrow$SU{state($a_i$,abort)}
      else // max delay of the parents
        $\Delta$=max{ $\Delta_x$|x$\in$pa($a_i$,G),state(x,delay($\Delta_{x)}$$\in$S}
        S$\leftarrow$SU{state($a_i$, $\Delta$)}
     else if //action has been aborted
          (calltime($a_i$,L)=abort)
         then //set action state to abort
           S$\leftarrow$SU{state($a_i$,abort)}
         else //compute delay
           $\Delta$=$\lceil$ calltime($a_i$,L)- eventtime($a_i$,L)$\rceil$
   for i=1 to n do
    if (state($a_i$,abort)$\in$ S)
     then eventtime($a_i$,L)$\leftarrow$abort
    if (state($a_i$,delay($\Delta$))$\in$ S)
     then eventtime($a_i$,L)$\leftarrow$ eventtime($a_i$,L)+$\Delta$
   return L'$\leftarrow$
   end.
```

The plan reconfiguration algorithm PR takes the plan causal graph and the execution state and outputs an updated checklist of the plan. A plan execution state consists of a determination in real-time for each event, relative to its call time and scheduled time, whether: the event is late; has not occurred; been aborted; called in early; called in on time; called in late. Our reconfiguration algorithm first orders the events along a causal order which is just a topological sort of the causal graph (which is acyclic). The algorithm propagates the execution state by aborting an event if any of its parents is aborted; otherwise delaying the event by the maximum delay over its parents. The algorithm shifts the scheduled time for delayed events and outputs a new execution list ordering the events by increasing event time. Figure 4 shows a reconfiguration example for our SOF plan of Figure 2, where the depart event of the fire support gunship (AC-130U) is delayed 1 hour. The reconfiguration is computed based on the amended causal graph shown in Figure 3. Compared to the original plan in Figure 2, we see that (a) the event of KC-135 at air-refueling track is shifted back 1 hour; (b) the departure of the chinook from Copper and the Blackhawk from Ship1 are all shifted by same amount to meet the fire-support coverage constraint; (c) all causally dependent events are properly shifted; (d) all independent events remain unchanged, e.g. the decision point the departure of the Chinook from baseX and arrival to Copper. Note that the end times for places are shifted by proper amount to satisfy the condition that reconfigured place events be within the start and end time for the place.

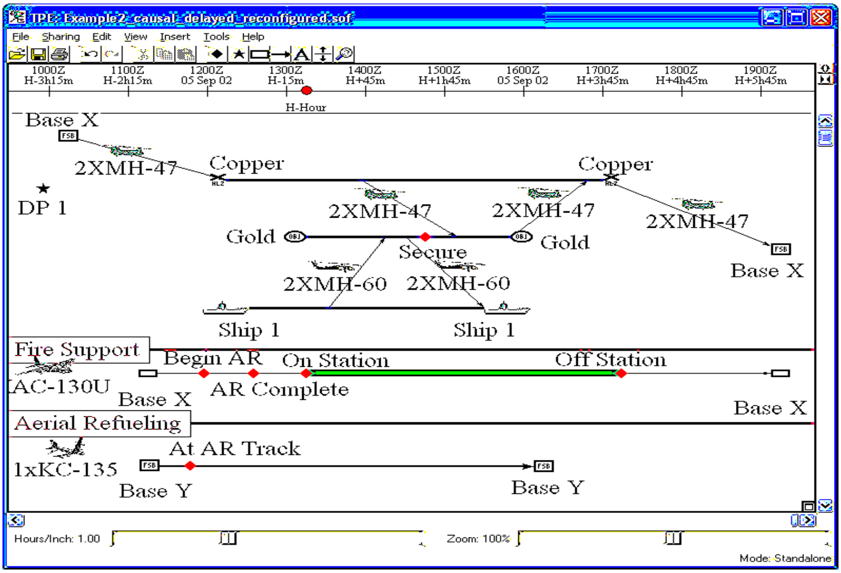


Fig. 4. Reconfigured plan after fire support delayed 1 hour.

4 Conclusion

The paper presents an approach for computing parsimonious representation of causal dependencies in a plan authored directly by humans without requiring complete domain models. The paper computes a minimal DAG whose nodes are the actions and the directed arcs represent potential causal links and/or threat resolution links as in partial order planning. The term potential emphasizes the uncertainty in the abduced causal relation since no requirement is placed on the availability of complete domain theory. The paper describes an algorithm that computes the minimal potential causality DAG based only on the temporal ordering of the actions of the input plan and on the task decomposition tree associated with the plan. The DAG is presented to the plan author for validation who can then amend the potential causality structure graphically by adding or deleting directed arcs between the action nodes. The validated causal graph captures the knowledge relevant to the causal dependencies in the plan and can then be used for reasoning about plan reconfiguration in the context of execution. Our approach has been implemented and used in a working prototype in a real life Special Operation Forces (SOF) planning domain. We regard the present results as preliminary and plan to extend the investigation in various directions. We will investigate other planning domains in the context of MIP and extend the approach to take advantage of partial domain theories when available. It will be interesting to evaluate empirically the accuracy of the potential causality relation by comparing with partial order planners for artificial domains having complete models. We will investigate a formalization of the tradeoff between accuracy of the potential causality relation and the completeness of the description of the planning domain.. We will also integrate our approach with other techniques in case-based planning and qualitative reasoning to enhance the plan authoring process and to enable more effective techniques for replanning and dynamic execution.

References

1. Weld, D.: Recent advances in AI planning. AI Magazine 20 (1999) 93–123.
2. Burstein, M., McDermott, D.: Issues in the development of human-computer mixed-initiative planning, Cognitive Technology, B. Gorayska and J.L. Mey (eds.), Elsevier (1996) 285-303.
3. Kambhampati, S., Kedar S.: A unified framework for explanation-based generalization of partially ordered and partially instantiated plans, Artificial Intelligence, 67 (1994) 29-70.
4. Veloso, M., Carbonell J.: Derivational analogy in prodigy: Automating case acquisition, storage and utilization. Machine Learning (1993) 249–278.
5. Veloso, M.. Towards mixed-initiative rationale-supported planning. In Austin Tate, editor, Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative,. AAAI Press (1996) 277-282.
6. Myers, K.L.: Toward a theory of qualitative reasoning about plans. Technical report, A.I. Center, SRI (2001).
7. Veloso, M., Perez, M., Carbonell, J.: Nonlinear planning with parallel resource allocation. In Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control (1990) 207-212.
8. Regnier, P., Fade, B.: Complete determination of parallel actions and temporal optimization in linear plans of action. In Joachim Hertzberg, editor, European Workshop on Planning, volume 522 of Lecture Notes in Artificial Intelligence, Springer (1991) 100-111.
9. Bäckström, C.: Finding Least Constrained Plans and Optimal Parallel Executions is Harderthan We Thought. In Proceedings of the Second European Workshop on Planning (1993).

Reactive Planning Simulation in Dynamic Environments with *VirtualRobot*

Oscar Sapena[1], Eva Onaindía[2], Martín Mellado[3], Carlos Correcher[3] *, and Eduardo Vendrell[3]

[1] Dept. de Ciencias de la Computación e Inteligencia Artificial, Univ. Alicante, Spain
osapena@dccia.ua.es

[2] Dept. Sistemas Informáticos y Computación, Univ. Politécnica Valencia, Spain
onaindia@dsic.upv.es

[3] Dept. Ingeniería de Sistemas y Automática, Univ. Politécnica Valencia, Spain
{martin, even}@isa.upv.es, carlos.correcher@jrc.it

Abstract. This paper describes the architecture of a reactive planning system for dynamic environments, which is specifically designed to deal with robot planning problems. The architecture permits many agents to work simultaneously on the same environment and it is aimed at working with incomplete information. Agents have partial knowledge about the world and data soon becomes obsolete because of the changes in the environment. Our approach is designed to overcome this difficulty through a highly coupled system composed of an incremental planner and an executor. The whole system is integrated into *VirtualRobot*, a graphical software application, which provides a flexible and open platform to work on robotics. Through *VirtualRobot* we can incorporate important features into the system as simulation of sensing actions or a monitoring mechanism. Additionally, the planning algorithm is able to work in time-limited situations and use numeric variables. All these features make our planning system be a nice toolkit to deal with reactive robot planning.

1 Introduction

The problem of classical planning involves generating a sequence of actions which, applied to an initial state, allows to achieve a set of goals. Research in classical planning has been carried out under some non-realistic assumptions as static, deterministic and completely accessible environments [12]. In order to overcome these simplifications, new approaches like universal, conformant, conditional or probabilistic planning have arisen. Nevertheless, in these approaches, the plan execution monitoring is just a simple mechanism that executes the received plan and checks that everything happens as planned. In case that an unexpected event occurs, it is necessary to compute a new plan from scratch or to try to repair the old plan [2].

In general, plan generation for autonomous systems (like mobile robots) cannot be separated from its execution. Planned actions will be executed in unpredictable and

[*] Currently working in Institute for the Protection and Security of the Citizen (IPSC), Joint Research Centre – ISPRA.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 699-707, 2004.

dynamic environments, so plans are more likely to fail. Moreover, there is information that can only be acquired during execution time. Reactive planning integrates plan generation and execution, which constitutes a suitable platform to deal with real-world problems. In reactive planning, an agent is defined as a combination of a planner plus a reactor [13], and this approach is the basis of the work presented in this paper.

The integration of the planner and the reactor is carried out through the use of *VirtualRobot Simulator* (*VRS)*, a graphical simulator part of *VirtualRobot* suite. Nowadays, graphic simulators for robotic systems are indispensable in most of the robot design, learning and exploitation steps. Technological advances and improvements in computing engineering allow this kind of applications to be applied on any field in robotics: industrial robotics [3], mobile robotics [14], sub-aquatic robotics [1] or aerospatial robotics [7]. In addition, they have become significantly more powerful and flexible. Thanks to the inclusion of new capabilities such as sensor data, simulation software is not only dedicated to simulate robot behaviour, but also is useful for design, analysis and validation of techniques as collision detection, motion planning, sensor modelling, evaluation of control architectures and so on.

VirtualRobot has been originally created for remote robot monitoring, programming and simulation of the robot control system *GENERIS*[1] [11], but has become a useful general tool in many fields of robotics, such as manipulator-robot programming, walking robot simulation, mobile-robot control, distance computation, sensor simulation, collision detection, motion planning and so on.

2 System Architecture

The system is composed of several agents working in the same environment. Agents can be classified in planning agents and external agents. Agents of the former type are composed of a planner and a reactor (see Fig. 1). They are in charge of computing and executing plans to achieve goals, which can vary along the time. External agents, like people or natural phenomena, are capable of modifying the environment without the need of the knowledge of the planning agents. They produce most of the unexpected events that the planning agent detects during the plan execution through a monitoring process.

Initially, planning agents have some knowledge about the state of the world (environment model). Normally, this knowledge will be incomplete and inaccurate. The reactor asks the planner for one or more actions when it is required in the domain. If there are no deadlines to return a plan, the answer can be delayed until the planner has computed a good plan. When the reactor requests an action, the planner must reply as soon as possible according to the environment time constraints. The reactor translates the action into a set of low-level actions (see Example 1) and it sends these actions to the environment. Therefore, a high-level action cannot be considered as an atomic executable action, so the reactor must have a recovery mechanism to reach a valid state if the action execution fails in an intermediate stage. In this case, the reactor updates

[1] Generalised Software Control System for Industrial Robots, developed by European Commission Joint Research Centre

the information the planner has about the world and notifies the unexpected outcome to the planner.

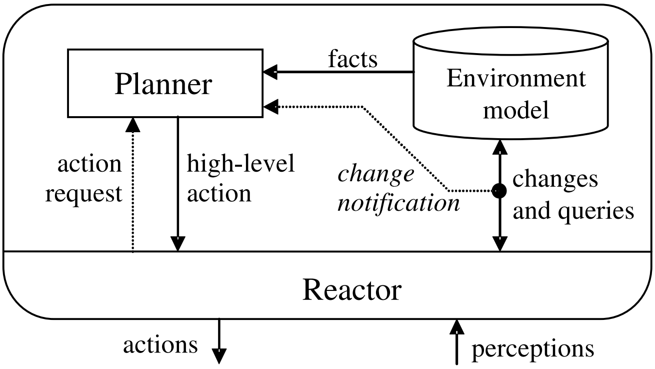


Fig. 1. A planning agent is composed of a planner and a reactor.

Example 1. One high-level action (MOVE rob A B) for moving a differential robot rob from room A to room B can be translated into the following sequence of low-level actions:

```
compute_angle α between A and B;
if (battery_available) then rotate rob α degrees;
while (battery_available) and (not collision) and
      (position(rob) ≠ B) do one_step_forward rob;
end while;
```

The graphical simulation of the execution is carried out through the *VRS* application. At initialization time the system input are three parameters: a domain description, a problem description and the environment description. The system is able to work under any robotics graphical simulator so in the environment description it must be specified that *VRS* is the toolkit to be used. The integration with *VRS* is done through a dynamic link to a library. The library has to receive the following information during its initialization:

- The domain name.
- The environment description. This description is provided in a separate file, which contains information about the geometry, colors and textures of the static objects (objects that cannot be moved or altered) in the environment.
- The problem objects: each object has a name, a type, some coordinates to locate the object in the environment and a physical model (containing geometric and kinematics parameters). *VRS* distinguishes two types of objects: parts - objects that can be handled - and crafts - objects that can modify the environment.

Once the environment is initialized, the system creates the agents. Planning agents receive the domain and problem description as well as the library to communicate with the environment. The planner uses the information from the problem and domain to compute the plans. The reactor, which is domain-dependent, is implemented into the library. The communication between the reactor and the planner is established through two callback functions:

- ***Environment_action_request***: this function is used by the reactor to request the planner for an action.
- ***Environment_monitor_info***: this function is used by the reactor to inform the planner about unexpected events.

When the planner receives an action request, the process of generating plans is halted and the planner sends an action to the reactor through the function ***SendAction***. The reactor translates this high-level action into primitive actions, which are directly executable in the environment and are understandable by *VRS*. If the action execution fails, the reactor must be able to reach a high-level state, i.e. a valid planning state. Then, the reactor communicates the planner the unexpected changes in the environment. If these changes affect the calculated plans, the planner restarts the computation of new plans and discards the old ones (for the moment, the planner does not use any replanning technique to reuse the old plans).

3 Reactive Planning

In a reactive environment it is not feasible to find a complete plan before starting the execution. That is the reason why the planning algorithm follows the design principles of the anytime algorithms [2]. The planning algorithm is also based on the divide-and-conquer methodology: split the problem into smaller subproblems, solve these subproblems and combine the obtained solutions. The planning algorithm computes a plan for each top-level goal separately. However, combining these plans may cause conflicts between each other that are hard to solve [15].

The overall working scheme is shown in Fig. 2. A planning problem $P=(O,I,G)$ is a triple where O is the set of operators, I the initial state and G the top-level goals. This algorithm starts from the current state S_0, which initially corresponds to I. The planning algorithm works in four stages:

- **The preprocess stage**. This stage consists of computing all ground actions and literals, starting from the operators, predicates and objects. These calculations speed up the following planning stages. This preprocessing stage is usually done once but, in dynamic environments, the number of actions and literals can change, so new objects can appear giving rise to new possible actions and literals.
- **Calculation of the initial plans**. An incomplete plan is computed for each non-achieved goal $g_i \ / \ g_i \in G \ \wedge \ g_i \notin S_0$. Therefore, P is decomposed in n planning subproblems $P_1=(O,S_0,g_1)$, $P_2=(O,S_0,g_2)$,..., $P_n=(O,S_0,g_n)$, where n is the number of non-achieved goals. These plans are computed starting from a relaxed planning graph (where delete effects of actions are ignored) [4], and they are generated very rapidly (polynomial time). We must remark that these plans are not completely executable in most cases. However, these initial plans are aimed to be used as a starting point for further refinements.

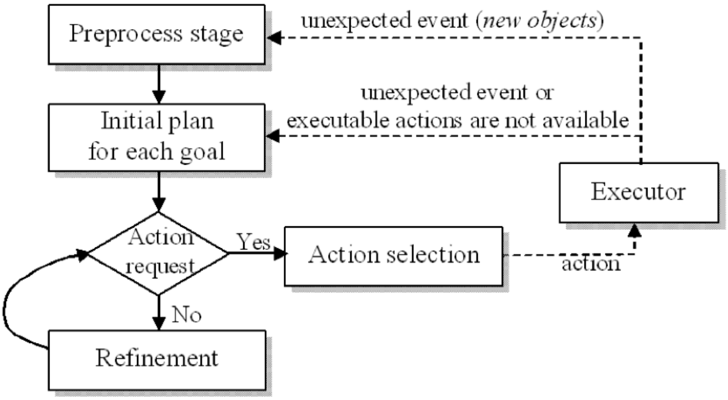


Fig. 2. Outline of the planning algorithm.

- **Refinement of the plans**. While the reactor does not require actions to be executed, the planner proceeds with the refinement of the incomplete plans. This refinement continues until all plans are completed, an unexpected event invalidates the current plans or the available time expires. The refinement stage repairs the plans through the insertion of new actions to make the plan completely executable. Each refinement stage attempts to solve a non-solved literal in each plan.
- **Selection of the action to be executed**. When the reactor requests for an action, plans are ordered according to a conflict checking criteria [12]. This way, a plan P_i is ordered before a plan P_j when it is necessary to execute the first action of P_i before starting the execution of P_j. The next action to be executed will be the initial action of the plan ordered in the first place.

4 Simulation Platform: *VirtualRobot*

VirtualRobot [8][9] is a freeware software suite[2] in the sense that includes several programs for robotics application, research and education, with a graphical representation based on *OpenGL*. *VirtualRobot* is designed for low cost platforms and used as the common interface for all the applications. *VirtualRobot* is composed of:

- A basic geometric modeller *VirtualRobot Modeller* (*VRM*), to create and edit geometric and kinematics models.
- A geometric data translator, *VirtualRobot Translator* (*VRT*) to convert files from *AutoCAD® DXF* and *VRML* file formats in addition to a special plug-in module to export data from *3DStudio Max®*.
- The main platform for simulation, *VirtualRobot Simulator* (*VRS*), including some components (a set of adaptable Dynamic Link Libraries) and external applications (*VRS Tools*, *VRM Tools* and *VRS Demos*).

[2] *VirtualRobot* can be downloaded in http://www.isa.upv.es/~vrs

VRS can be applied for simulating any type of robots, individually or grouped in multi-robot workcells for their off-line program generation and testing, as well as for on-line programming and monitoring. *VRS* can be connected to the numerical control *GENERIS* using *TCP/IP* to monitor the process and robots status. *VRS* can simulate any kind of robots, that is, not only manipulator-robots, but also multi-axis machines as conveyors, turntables, machine-tools, sensor systems and crafts (general mobile-robots). Robots can be attached one to one, or many to one, in order to form more complex and redundant devices, such as walking robots.

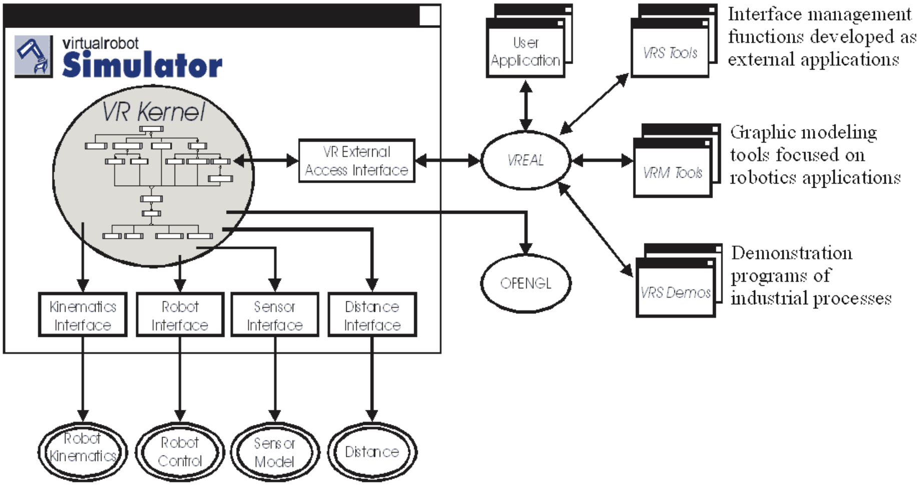


Fig. 3. *VRS* Software architecture.

4.1 *VRS* Software Architecture

VRS software architecture (Fig. 3) is based on the following four parts:

1) *VRS Kernel*, including graphical display control, objects data designed through a hierarchical structure of classes following the inclusion principle and processing threads. *VRS Kernel* handles user action events and external application orders.
2) A set of external components under *VRS* that the user can replace with his/her own components in such a way that *VRS* can be adapted to the user requirements. The external components are Dynamic Link Libraries (*dlls*) loaded on memory during execution time. Robot kinematics, sensor models and distance computation are example of external components.
3) An external access library (*VReal)* for external application development. *VReal* is implemented as a Dynamic Link Library in order to make possible the interaction among client's application and *VRS*.
4) The external applications that run over *VRS*. The external applications form the real user interface of any specific application. Once again, the user can adapt *VRS* to special requirements, constructing the required interface as new external applications.

4.2 Planning Domain Models in *VirtualRobot*

With the help of *VirtualRobot*, typical planning domains can be modelled in order to simulate the behaviour of planning techniques. We can consider the following three representative planning domains:

- **Robot-part domains**: in this type of domains, manipulator-robots grasp and move objects, that is manipulate movable objects. Typical robot-part domains are the *Blocksworld* domain [6] and *Hanoi* domain [6]. In *VRS* it is possible to load manipulator-robots from a large library of robots and new robots can also be modelled. Parts can also be easily modelled in *VRM* specifying its geometry and grasping frames. *VRS* features for robot programming include object-oriented orders in such a way that picking and placing parts can be done in a very intuitive way.
- **Mobile-robot domains**: in this type of domains, mobile-robots navigate around their environments, searching for a goal, while avoiding obstacles (Fig. 4). Problems of transportation, like the classic planning domains *Logistics* [6] (for trucks), *Ferry* [6] (for ships) or *Satellite* [7] (for aero-spatial vehicles) are examples of mobile-robot domains. A library of characteristic crafts and their corresponding kinematics components are included within *VirtualRobot* but new crafts can be easily implemented. External applications have access through speed commands to the craft speed control in order to guide craft motion. Robot navigation in mazes or part dispatching are representative tasks in these domains. For navigation, crafts can include distance and reflector sensors.
- **Mix domains**: this type of domains combines manipulator-robots and mobile-robots to be applied on complex tasks. A typical planning domain of this kind is *Depots* [7] (with trucks and hoists). In *VRS*, robots can be attached in order to form complex devices (Fig. 5). Coordination between different robots is possible by means of different functions that establish digital and analog connections between robots.

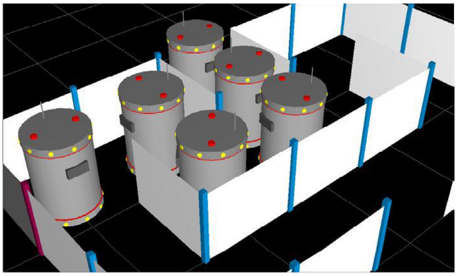


Fig. 4. A mix domain in *VRS* with a manipulator-robot attached to a tricycle wheeled robot.

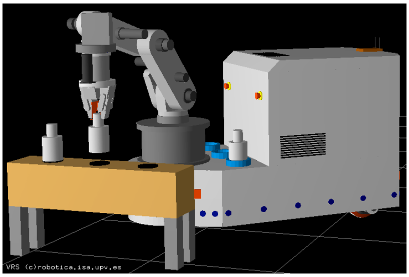


Fig. 5. Simulation of a differential wheeled robot in *VRS*.

5 Results

The standard way of evaluating the performance of planners is by comparison with other state-of-the-art planners. In classical planning, comparisons are easy since the world is assumed to be static, deterministic and accessible (closed world assumption). When the world does not comply with these simplifications, comparisons between planners are not trivial tasks. The planning community is now conscious of the new demands and, there exist several proposals to deal with more realistic assumptions. In order to do an assessment of the performance of on-line planners, they are executed with simulated scenarios. Planning quality is evaluated as a combination of the cost of the plan (referred to the problem metric function) and the running time.

Table 1 shows some preliminary results to give information about the quality and computational cost of the planning process. The table shows the plan length (in number of actions) and the running time (in milliseconds) for solving several different problems in the *Blocksworld* [6] and *Satellite* [7] domains. In *Blocksworld* domain problems, a robot arm must pick up and stack blocks in order to obtain one or more block piles in the right order. In the *Satellite* domain, it is necessary to set up several instruments and to turn the satellites to the right direction in order to take photographs of some astronomical phenomena. Results show that the planner is able to compute plans very rapidly. Moreover, the computational cost depends on the length of the final plan rather than the problem complexity (number of objects, literals and actions), although both facts are usually interrelated. The plan length is compared to the best solution found by the planners in the competitions, since none of the planners can guarantee the optimal plan . In general, the quality of the obtained plans are very close to the best available solutions.

Table 1. Preliminary results obtained. The planner has been executed in a *K7 – 1.4 Ghz*.

	P1	P2	P3	P4	P5	P6	P7
Blocksworld	5 blocks	7 blocks	9 blocks	11 blocks	13 blocks	15 blocks	17 blocks
Plan length/Best	12/12	20/20	30/30	34/32	42/42	48/40	54/46
Time (ms.)	5	10	10	20	30	60	110
Satellite	1 sat.	2 sat.	3 sat.	4 sat.	5 sat.	6 sat.	10 sat.
Plan length/Best	9/9	13/11	20/16	26/26	35/32	42/41	48/48
Time (ms.)	5	5	40	180	350	1000	1460

6 Conclusions and Future Work

In this paper we have described a system for planning in dynamic environments and with incomplete information. The architecture of the system allows many agents to work simultaneously on the same environment and, therefore, it is possible to check the behavior of the planning agents when unexpected events occur. Moreover, the system includes some other useful features to work in real-world domains: support for sensing actions, execution monitoring, planning in time-limited situations and use of numeric variables. The environment is graphically simulated with the *VirtualRobot* toolkit, which has shown its great possibilities for this kind of problems. The architecture and integration of *VRS* with the planner have been described in the paper.

The planning algorithm is based on the divide-and-conquer methodology and computes a plan for each top-level goal separately. This technique allows to tackle quite large problems without an excessive computational effort. However, the planning algorithm is not complete but, in practice, some preliminary results show that the obtained plans are very close to the optimal ones.

There is still a lot of work to be done. It is necessary to develop simulations for some of the well-known classical planning domains, and find some new domains that can take advantage of the new system features (there are very few benchmark suites that include sensing actions and uncertainty). The system can also be improved to provide the reactor with several parallel actions in order to exploit the inherent parallelism of the real world (for example, when handling several robots at the same time). Another interesting work would be the extension of the planning algorithm to support probabilistic domains. Probabilistic domains provide the planner with additional information and help the planner choose the alternatives that are more likely to succeed. The feature of *VirtualRobot* allows to implement all these problems very easily.

Acknowledgements. This work has been partially funded by projects MCyT TIC2002-04146-C05-04, FEDER-CICYT DPI2001-2094-C03-03, DPI2002-04434-C04-04 and UPV 20020647, 20020681.

References

1. Chen, X., Marco, D., Smith, S., An, E., Ganesan, K., Healey, T.: 6 DOF Nonlinear AUV Simulation Toolbox. IEEE, (1997).
2. Drummond, M., Swanson, K., Bresina, J., Levinson, R.: Reaction-first Search. In Proceedings of the IJCAI-93. (1993) 1408-1414.
3. Freund, E., Rokossa, D., Rossman, J.: Intuitive Off-line Programming of Industrial Robots Using VR-Techniques. 15th ISPE/IEE Int. Conf. on CAD/CAM, Robotics and Factories of the Future. (1999).
4. Hoffman, J., Nebel, B.: The FF Planning System: Fast Planning Generation Through Heuristic Search. In JAIR. (2001) 14, 253-302.
5. International Planning Competition (2000). *http://www.cs.toronto.edu/aips2000*.
6. International Planning Competition (2002). *http://www.dur.ac.uk/d.p.long/competition.html*.
7. Kazuda, Y.: Experimental Study on the Dynamics and Control of a Space Robot with Experimental Free-Floating Robot Satellite (EFFORTS) Simulators. Advanced Robotics. (1995).
8. Mellado, M.: Simulación en Robótica Mediante *Virtual Robot*. Universidad Politécnica de Valencia. Ref.: 2003.443. (2003).
9. Mellado, M., Correcher, C., Catret, J.V., Puig, D.: *VirtualRobot*: An open general-purpose simulation tool for robotics. Eurosis International Conference ESMc. (2003).
10. Pollack, M.E., Horty, J.F.: There's More to Life than Making Plans: Plan Management in Dynamic, Multi-Agent Environments. AI Magazine. (1999) 20(4), 71-84.
11. Ruiz, E.: GENERIS: The EC- JRC Generalised Software Control System for Industrial Robots. Int. Journal of Industrial Robot. (1999) 26(1).
12. Sapena, O., Onaindia, E.: A Planning and Monitoring System for Dynamic Environments. Journal of Intelligent and Fuzzy Systems. (2002) 12(3-4), 151-162.
13. Wolverton, M., Washington, R.: Segmenting Reactions to Improve the Behavior of a Planning/Reacting Agent. In Proc. of AIPS. (1996) 245-250.
14. Yang, L., Yang, X., He, K., Guo, M., Zhang, B.: Research on Mobile Robot Simulation & Visualization. IEEE Int. Conf. on Systems, Man and Cybernetics. Piscataway. (1997).
15. Yang, Q.: Intelligent Planning. A Decomposition and Abstraction Based Approach. Spinger-Verlag. Berlin, Heidelberg, (1997).

New Distributed Filtering-Consistency Approach to General Networks

Ahlem Ben Hassine[1,2], Khaled Ghedira[2], and Tu Bao Ho[1]

[1] School of Knowledge Science, Laboratory Knowledge Creation Methodology,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai Tatsunokuchi-machi, Nomi-gun, Ishikawa, 923-1292 Japan
{hassine, bao}@jaist.ac.jp
http://www.jaist.ac.jp/ks/labs/ho/

[2] Laboratory SOIE, High Institute of Management
41, Avenue Liberte- Cite-Bouchoucha, le Bardo, Tunisia
khaled.ghedira@isg.rnu.tn

Abstract. Reinforcing local consistency, in particular arc consistency, has attracted the attention of many researchers due to its preeminent role in efficient CSP (constraint satisfaction problem) solving. Many centralized arc consistency techniques have been proposed for *binary* constraints. But, only very few reports are available addressing directly to general constraints for centralized framework and not for distributed one. The main contribution of this work consists in proposing a new generalized schema (called GDRAC) for reinforcing arc-consistency on general constraint networks (CNs) in an asynchronous and distributed manner. The experimental comparative evaluations show the high performance of our approach to general constraints.

1 Introduction

Constraint satisfaction problems (CSPs) formalism is widely used in many real world applications (e.g. planning, scheduling, resource allocation, etc.). The great success of the CSP paradigm is due to its simplicity. In this manner, to model a problem as a constraint satisfaction problem, we specify a search space using a set of variables, each of which can be assigned a value from some finite domain of values. This model includes constraints that restrict the set of acceptable assignments.

The task of solving CSP is NP-hard, and much research effort has been concentrated on improving the process of finding solution for this problem. Reinforcing local consistency and especially, the most preeminent one, reinforcing arc consistency caught many researchers' interest. This framework is marked off by the ubiquitous presence of many works. Nevertheless, the majority of them deal with binary problems, i.e. problems where each constraint involves at most two variables. The main reason is that any non-binary problem can be transformed into a binary one. For this reason many methods are proposed in literature to translate a non-binary constraints into an equivalent set of binary ones. However, in [14] the author proved that this transformation could lead to a loss of a part of the constraints' semantics.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 708-717, 2004.

Recently more attention has been turned to non-binary problems. Few works dealing with reinforcing arc-consistency for non-binary problems have been proposed in the literature. CN proposed by [10] is a generalization of AC-3 [9] to non-binary constraints. The worst time complexity of this algorithm is $O(er^2d^{r+1})$, where e is the number of constraints in the network, d is the size of the largest domain and r is the maximal arity of the constraints. Another algorithm GAC-4 was proposed by [13] uses the same idea as AC-4 [12]. Its time complexity in the worst case is $O(ed^r)$. According to [4], CN can only be applied to ternary constraints and very small domains, while GAC-4 can only be applied to very tight constraints, where the number of allowed tuples of values is very small. [14] suggested an efficient algorithm for enforcing generalized arc-consistency on a set of *all-different* constraints. Bessiere and Regin [4] have proposed a generalized schema to perform arc-consistency on any real constraint network GAC-schema. This schema is based on the AC-7 algorithm given in [2, 3]. It uses the ideas of "current support" and of "multidirectionality", i.e. the generalization of the "bidirectionality" property to the non-binary constraint. The worst space complexity of this algorithm is $O(r^2d)$, while its time complexity is $O(d^r)$.

All these works are addressed to the centralized framework, and no work has been proposed on the distributed framework. But with the natural distribution of many real CSP applications and the advents of both distributed computing and networking technologies, it is time to focus our research on distributed approaches. Our main contribution in this paper is to propose a new generalized schema (named as *G-DRAC*) for reinforcing arc-consistency on any generalized constraint network. The basis of the new approach is the DRAC approach [5; 6] in view of its advantages. The main objective of this later approach is to obtain the full global arc consistency on binary CSPs as a result of interactions between a set of reactive agents. The use of dual constraint graph [7] in the DRAC model allows us to directly address generalized constraints without having any resorts to transform the initial problem into a binary one.

This paper is organized as follows. First we will present some useful definitions and notations. Next, we will explain the generalization of DRAC approach to any general CN. Then we will prove its complexity. Finally, we present and discuss the experimental evaluation and give our conclusion.

2 Definitions and Notations

In this section, we formally define *constraint satisfaction problem* formalism and give some useful notations and definitions.

A constraint satisfaction problem [11] is a tuple (X, D, C) where:

- X={X_1, …, X_n}, is a finite set of n variables,
- D={D(X_1), …, D(X_n) }, is a set of n finite domains. A total order $<_d$ can be defined on the values of each domain, without loss of generality.
- C={C_1, C_2, …, C_m} is a set of m constraints between these variables. Each constraint C_i implies an ordered set of X(C_i)=(X_{i_1}, …, X_{i_r}) variables. |X(C_i)|=r is the arity of the constraint. The constraints restrict the values the r variables can si-

multaneously take. If r =2, the constraint is called binary constraint, otherwise it is n-ary constraint. Thus, each constraint C_i is a subset of the Cartesian Product $D(X_{i_1})$x …x $D(X_{i_r})$ that specifies the allowed combination of values for the variables. An element $(a_{i_1}, a_{i_2}, \ldots, a_{i_k}, \ldots, a_{i_r})$ of $D(X_{i_1})$ x …x $D(X_{i_r})$ is called a *tuple* on $X(C_i)$. A tuple t on $X(C_i)$ is *valid* if for all $a_{i_k} \in t$, $a_{i_k} \in D(X_{i_k})$,

A constraint can be represented implicitly, where a computation is needed to answer constraint check questions, or explicitly where the answer is already recorded in a database. Solving a CSP requires finding one or all-complete assignments of values to all the variables of the problem, satisfying all the constraints.

Let P (X, D, C) be a CSP, given a constraint $C_j \in C$ and $X_{j_k} \in X(C_j)$. A value $a_{j_k} \in D(X_{j_k})$ has a support in C_j if and only if there is a tuple t that satisfies C_j and such that $t[index(C_j, X_{j_k})]=a_{j_k}$ ($index(C_j, X_{j_k})$ returns the index of X_{j_k} in C_j). t is then called the support of (X_{j_k}, a_{j_k}) in C_j.

A constraint C_j is arc-consistent if and only if each value a_{j_k} of each variable $X_{j_k} \in X(C_j)$ has a support in C_j. A CSP is arc consistent if and only if it has non-empty domains and each of its constraints is arc-consistent.

A non-binary CSP is a generalized arc-consistent (GAC) if and only if for any variable X_{j_k} in a constraint C_j ($X_{j_k} \in X(C_j)$) and value $a_{j_k} \in D(X_{j_k})$ that it is assigned; there exist compatible value $a_{j_l} \in D(X_{j_l})$ for all the other variables X_{j_l} ($X_{j_l} \in X(C_j)$) in C_j. Van Beek and Dechter [15] have proposed another definition of arc-consistency for non-binary constraint networks, namely relational arc-consistency. This definition requires global consistency on the sub-network formed by the variables of the constraint and all the other smaller constraints implying some of these variables. Reinforcing arc-consistency on a CSP consists on repeatedly removing unsupported values from the domain of its variables.

For non-binary constraint, Bessiere and Regin [4] have proposed the property of "multidirectionality" of constraints. It is defined by the fact that for any constraint C_j, a tuple t on $X(C_j)$ is a support for the value $t[index(C_j, X_{j_k})]$ where $X_{j_k} \in X(C_j)$ if and only if for all $X_{j_h} \in X(C_j)$, t is a support for $t[index(C_j, X_{j_h})]$. We say that an algorithm "deals with" multidirectionality if and only if it never checks whether a tuple is a support for a value when it has already been checked for another value, and never looks for a support for a value on a constraint C_j when a tuple supporting this value has already been checked.

In this section, we propose two new properties that we will use in the proposed protocol in order to decrease the number of constraint checks and consequently to improve the efficiency of our approach without loss of correctness. We should note that the main idea of these properties is to extract "more" knowledge from the "multidirectionality " property of constraints to increase the efficiency of the approach.

Property 1 (for binary constraint)**.** For each binary constraint C_j, for each candidate[1] value $a \in D(X_{j1})$, "hide" from the domain of the related variable $D(X_{j2})$ all the values b having as a *first* value support $a' \in D(X_{j1})$ such that $a' > a$, and vice versa.

A sketch of proof. We will simply show that each hidden value b is not compatible with the value a. Therefore, we suppose that $\exists\ t' \in C_j$ such that $t'[index(C_j, X_{j1})] = a$ and $t'[index(C_j, X_{j2})] = b$. If this tuple exists then $t' \prec_{lo} t$ ($t \in C_j$ such that $t[index(C_j, X_{j1})] = a'$ and $t[index(C_j, X_{j2})] = b$) because $a' > a$. So t cannot be the *first* tuple support of b.

Property 2 (generalization for n-ary constraints). For each n-ary constraint C_j, for each candidate value $a_{jk} \in D(X_{jk})$, "hide" from the domains of All the related variables X_{jh}, $X_{jh} \in X(C_j)$ and $h \neq k$, all the values a_{jh} such that:

1. t is the *first* tuple support a_{jh} in C_j such that $t[|X(C_j)|+1] = h-1$, $t[\text{index}(C_j, X_{jh})] = a_{jh}$, $t[\text{index}(C_j, X_{jk})] = a'_{jk}$ and $a'_{jk} > a_{jk}$,
2. $\forall\ a_{jl} \in \boldsymbol{t}$; $l \in \{1..\ |X(C_j)|\}$ such that $l \neq h$ and $l < k$, $a_{jl} = Last(D(X_{jl}))$.

A sketch of proof. For the first part of this property 1) the proof is the same as for the first property. However, the second condition 2) is added in order to guarantee that t is the highest tuple in C_j and not tuple t', that contains $t'[\text{index}(C_j, X_{jk})] = a_{jk}$, $t'[\text{index}(C_j, X_{jh})] = a_{jh}$ and $t \prec_{lo} t'$, exists.

3 DRAC Approach for General Network

G-DRAC approach is an adjustment of the DRAC approach to the general network. The underlying Multi-Agent model is the same as DRAC model, which consists of a set of Constraint agents related by the shared variables. Let us recall that two agents in our model are related by an edge if and only if they share at least one variable. All the agents in the system communicate with each other by exchanging point-to-point asynchronous messages. We assume that in our system we have two kinds of messages. The ordinary messages that should be received in the same order they were sent and the highest order messages that must be performed immediately when received. Any message is received in a finite delay.

In the following, we give first, the data structure of the proposed approach followed by the underlying global dynamic.

3.1 Data Structures

- $AcqConst^{C_j}[Y] = \{C_k\ /\ C_k \in C \text{ and } X(C_k) \cap X(C_j) = \{Y\}\}$, is the ordered set of all the Constraint agents sharing the variable Y with C_j.

[1] The value for which we are searching for a new *tuple* support.

- D^{Cj}={ $D^{Cj}(Y)$/ Y∈X(C_j)} represents the local view of the domains of all the variables implied in C_j. Each domain is supposed to be totally ordered. ∀ X_{jk}∈X(C_j), $D^{Cj}(X_{jk})$ is called as the occurrence of D(X_{jk}).

  Note that some occurrences of a given D(X_{jk}) may be different, but all occurrences of D(X_{jk}), ∀ j_k∈ {1.. *n*} must be identical when the full global arc-consistency is reached. At this step, let us refer to the final obtained domain $D^{Cj}(X_{jk})$ by $f\,D^{Cj}(X_{jk})$.
- *TupleSupport*$_{C_j}$ is the set of tuple (a_{j1}, a_{j2}, ..., a_{ji}, ..., a_{jr} ***y***) where ***r*** is the arity of C_j, and (a_{j1}, a_{j2}, ..., a_{ji}, ..., a_{jr}) satisfies C_j. The parameter *y*∈{0, 1, ..., *r-1*} is used to indicate that (a_{j1}, a_{j2}, ..., a_{ji}, ..., a_{jr}) is the "first" tuple support for $a_{j(y+1)}$. We suppose a natural lexicographic order $\prec_{lo}$ between tuples such that for all {*t*, *t'* }∈C_j, $t\prec_{lo}t'$ if and only if it exists *k* such that *t*[1..*k*-1]=*t'*[1..*k*-1] and *t*[*k*]< *t'*[*k*].
- *IncValue*^{Cj}[X_{jk}]={a_{jk} / ∄*t*∈C_j , *t*[index(C_i, X_{jk})] = a_{jk} and *t* is valid} represents the set of all the current inconsistent values for X(C_j).
- *HD*^{Cj}[X_{jk}]={a_{jk}∈ $D^{Cj}(X_{jk})$, ∀X_{jk}∈X(C_j) / a_{jk} verifies property 2} represent the new current domains after hiding some values.
- *ReviseValue*^{Cj} is the set of all the current values that should be revised.

3.2 Global Dynamic

The main objective is to transform a CSP *P* (X, D, C) into another CSP *P'*(X, D', C) equivalent via the interactions between the Constraint agents, which are trying to reduce their domains. At the initial state, the Interface agent creates all the Constraint agents and activates them (fig.1). Each agent C_j reduces the domains (D^{Cj}) of its own variables, i.e. ∀*k* ∈ {1.. *r*}, *r*=|X(C_j)| and X_{jk}∈X(C_j), by computing local viable values for each variable (Fig.1 lines 1 to 4). For achieving this, C_j looks for one *tuple support* (the first one) for each value of its variables. When the first support *t*, that satisfies C_j and *t*[Index(C_j, X_{jk})]∈$D^{Cj}(X_{jk})$, is found, then (a_{j1}, a_{j2}, ..., a_{jk}, ..., a_{jr} (*k-1*)) is added to the list of tuple supports *TupleSupport*$_{C_j}$. We must note that a_{j1}, a_{j2}, ...,a_{jk}, ..., a_{jr} are the first values support for a_{jk} but they are also values support for each other by applying the *multidirectionality* property of constraints relations (§ 2). A value a_{jk} is deleted from $D^{Cj}(X_{jk})$ if and only if it has no viable tuple support. Each obtained set of deleted values for a variable should be announced immediately to the concerned acquaintances in order to save fruitless consistency checks for these values.

Each agent that has received this message starts processing it by updating the domains of its variables by deleting non-viable received values (Fig.2 line 1). At the end of this computation, it updates computed support information's by deleting all non-viable tuples (Fig.2 line 3). In the case where a_{i_k} is an inconsistent value, the agent determines first all the tuple *t* / *t*[Index(C_j, X_{jk})]= a_{i_k}. Then tried to check the existence of another viable tuple support *t'* in *TupleSupport*$_{C_j}$ for each value a_{i_h}∈*t* (a_{i_h}∈ $D^{Cj}(X_{jh})$ and *k* ≠ *h*). In the negative case, the agent starts first by "hiding", from the domains of all the related variables (Fig.2 line 10), all the values that are incompatible[2] with a_{i_k} by using the aforementioned Property 2. This allows us to reduce the number of constraint checks. Second, it looks for another tuple support for each value a_{i_h} according to y and using the new domains. If y=(h-1) then the search must be done from the *smallest* tuple t' (according to the predefined order) such that $t \prec_{lo} t'$ (as AC-6). Otherwise, it looks for a support from the scratch i.e. the first (smallest) tuple in C_j. This can lead to a new values deletion and by consequence to new out coming messages. So reducing domains on an agent may, consequently, cause an eventual domain's reductions on another agent. Therefore, these interactions must carry on until the stable equilibrium state, where all the agents are definitely satisfied and consequently no more reduction is possible.

```
Begin
1.  For each $t \in C_j$ do
2.     For each $a_{j_k} \in t$ such that $a_{j_k} \in IncValue^{Cj}[X_{j_k}]$ do
3.        addTo ($TupleSupport_{Cj}$, ($a_{j_1}$, …, $a_{j_k}$ , …, $a_{j_r}$ ($k$-$1$)));
4.        Delete($IncValue^{Cj}$($X_{j_k}$), $a_{j_k}$);
5.  For each $X_{j_k} \in X(C_j)$ such that $k \in \{1, .., |X(C_j)|\}$ do
6.     Delete($D^{Cj}$($X_{j_k}$), $IncValue^{Cj}$[$X_{j_k}$]);
7.     If $D^{Cj}$($X_{j_k}$)=$\emptyset$ Then Send(Self, Interface, "StopBehavior"
8.  For each $X_{j_k} \in X(C_j)$ such that $IncValue^{Cj}[X_{j_k}] \neq \emptyset$ do
9.     For each $C_h \in AcqConst^{Cj}(X_{j_k})$ do
10.       Send($C_h$, self, "ReduceDomains:$IncValue^{Cj}$($X_{j_k}$) for: $X_{j_k}$")
End
```

Fig.1. Start message executed by each Constraint agent C_j.

An agent is satisfied when it has no more reduction to do on its variable domains or when one of its reduced domain wipes-out. But it is clear that this satisfaction state is not definitive. Indeed, if there exists at least one unsatisfied Agent, it may cause the unsatisfaction of other Constraint agents and this is due to the propagation of constraints. So, interactions and especially reductions must carry on.

Note that this dynamic allows a premature detection of failure: absence of solutions (Fig.1 line 7 and Fig.2 line 14). Thus, in the case of failure, the constraint (which has detected this failure) sends a message to the interface in order to stop the whole process. For thus, the Interface agent in turn send a message to each constraint to make them stopping their local activity and informs the user of the absence of solutions. The maximal reinforcement of global arc-consistency is obtained as a side effect from the interactions described above.

[2] This does not mean that the value is inconsistent.

The dynamic of DRAC approach stops when the system reaches its stable equilibrium state. At this state, all the agents are satisfied. An agent is satisfied when it has no more reductions to do on its variable domains or when one of its related new reduced domains is wiped-out. The detection of stable equilibrium state is achieved by using the well know algorithm of [8], state where all agents are waiting for a message and there is no message in the transmission channels. If all the agent of the system are in the state of waiting, and there exist only one agent C_j which has deleted one value *a* from the domain of one of its variables ($X(C_j)$). We assume that this agent shared this altered variable with another agent C_k. The agent C_k must be informed of the loss of the value *a* in order to propagate the constraints. Hence, there is a message in transit for it, which invalidates our transmission hypothesis.

```
ReduceDomain: DelVal for: Y
For each ($v \in DelVal$) such that($v \in D^{C_j}(Y)$) do Delete($D^{C_j}(Y), v$);
For each ($t \in TupleSupport_{C_j}$)such that $t[index(X(C_j), Y)] = v$ do
  Delete($TupleSupport_{C_j}$, $t$);
  For each $w \in t$ do
    If Last($t$)=(Index($t$, $w$) -1)
    Then AddTo($ReviseValue^{C_j}[index(t, w)]$, ($w$, $t$));
    Else AddTo($ReviseValue^{C_j}[index(t, w)]$, ($w$, $nil$));
For each ($w$, $t$)$\in ReviseValue^{C_j}$ do
  If (Check:$w$ for: Var($X(C_j)$), Index($t$, $w$))) = false
  Then $HD^{C_j} \leftarrow$ HideFrom: $D^{C_j}$ for:$w$ of:Var($X(C_j)$),index($t$, $w$));
      $t' \leftarrow$ SearchNewSupport:$w$ from:$t$ in: $HD^{C_j}$;
      If $t' = \varnothing$
      Then Delete($D^{C_j}$[Var($X(C_j)$), index($t$, $w$))], $w$);
         If $D^{C_j}$[Var($X(C_j)$), index($t$, $w$))]=$\varnothing$
         Then SendMsg(Self,Interface,"StopBehavior");
              AddTo($IncValue^{C_j}$(Var($X(C_j)$), index($t$, $w$))),$w$);
      Else AddTo($TupleSupport_{C_j}$, $t'$);
For each $X_{j_h} \in X(C_j)$ such that $IncValue^{C_j}[X_{j_k}] \neq \varnothing$ do
   For each $C_k \in AcqConst^{C_j}[X_{j_k}]$ do
      SendMsg($C_k$, self,"ReduceDomain:$IncValue^{C_j}[X_{j_k}]$ for: $X_{j_k}$");
End
```

Fig.2. Start message executed by each Constraint agent C_j.

4 Space and Time Analysis

Let a CSP *P* having *n* for total number of variables, *d* for the size of the variable domains and *e* for the total number of constraints and *r* for the maximal *arity* for each constraint. The number of Agents is *e*. If we consider a fully connected constraint network, we will have *e-1* acquaintances for each Constraint agent. Each agent C_j

maintains a list of tuples support $TupleSupport_{Cj}$. Each value should have at most one tuple support then the total number of tuples is at most rd. Each tuple has ($r+1$) values. Since there are $\boldsymbol{e}$ agents, the total amount of space for each agent is $(rd)(r+1)$. So the space needed for each agent, in the worst case, is O(r^2d), the same as GAC-Schema one's.

The worst case in the time execution of a distributed algorithm occurs when it proceeds with a sequential behavior. For our model, this occurs when only one value is deleted at a time. This leads to nd successive deletions. Our approach is composed of two steps; the first one is the initializing step, in which each agent performs d^r operations to generate the tuples support sets. For each deleted value, the agent will perform O(d^{r-1}) operations to search another tuple support for this value.

So the total time complexity of G-DRAC (with e agents and nd successive deletions), in the worst case, is O(end^r).

5 Experimental Evaluation

In this section, we provide experimental tests on the performance of the distributed filtering approach GDRAC. In our experimentation we have used GAC-7 [4] as a witness approach to appraise the fairness of the results of G-DRAC. The experimentations were performed over randomly generated problems using four parameters: n the number of variables, d the domain size of each variable, r the maximal arity of the constraints, p the graph connectivity (the proportion of constraint in the network, p=1 corresponds to the complete graph), q the constraint tightness (the proportion of allowed pairs of values in a constraint). The implementation was developed with Actalk, an object based on concurrent programming language with Smaltalk-80 environment.

We have randomly generated a list of problems according to the following parameters, n=20; d=10; r=3; $p \in$ {0.2; ...; 0.9} with step of 0.1 and $q \in$ {0.3; 0.38; 0.41; 0.43; 0.45; 0.46; 0.47; 0.48}. We have tried to carry our experimentation on problems, which belong to the transition phase, i.e. which consists on arc-consistent and inconsistent problems. For each <p,q>, 10 CNs instances were tested. Results reported below represent the average of the obtained results in mean of CPU time (in milliseconds) and constraint checks.

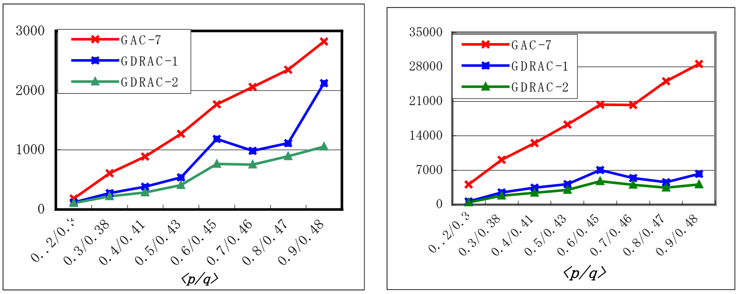


Fig 3. Results obtained in mean of CPU time (a) and number of constraint checks (b).

We have implemented two different versions of our approach. In the first version each agent follows out the received message after completing his current work, but in the second version each agent tries to execute each message immediately after receiving. Fig. 3 shows that for sparse CNs problems with loose constraints the both proposed approaches require almost the same CPU time. But the margin between the CPU times ruling in the two approaches increase with the density of the problems, i.e. the margin is more important for dense CNs with tie constraints. This can vindicate by the fact that informing agent by relevant information in time is very useful in order to save much fruitless constraint checks. As regards to GAC-7, it entails more CPU time than the two versions of our approach.

Concerning the constraint checks, the results above show that GDRAC-2 performs the minimum number. This advantage is due, partially, to the fact that the generated relations are expressed in extension, by a set of authorized values.

We tried also to carry on other measurements on the behavior of our approach towards inconsistent problems. The main objective of this second type of experimentation is to evaluate the percentage of deleted values requires for detecting the insolubility of a CN. Table 1 shows the ratio of the percentage of the obtained results of GDRAC-2 divided by that of GAC-7 in mean of deleted values and CPU time (in milliseconds). In the majority of cases, GDRAC-2 is a little bit more pruningful than GAC-7 to prove insolubility. Whilst in the other cases it is almost the same. At first glance, this result seems to be not good especially if this pruning process needs more time to be accomplished. However, the CPU time needed for our approach is less than that of GAC-7 (ratio<1). In the majority of cases GDRAC-2 requires around the half of time needed by GAC-7.

In addition, the difference in the percentage of the deleted values can be justified by the fact that for our approach all the constraints should be activated at the same time, leading to more deletions, even the problem is inconsistent. As for GAC-7 the insolubility of the problem can be detected in the beginning without invoking all the constraints.

Table 1. Results obtained in ratio of the mean of the percentage of deleted values and the CPU time

	<0.2; 0.2>	**<0.3; 0.3>**	**<0.4; 0.35>**	**<0.5; 0.4>**
% Deleted Values	1.05	0.86	1.03	1.15
CPU time	**0.46**	0.61	**0.46**	**0.52**
	<0.6; 0.4>	**<0.7; 0.42>**	**<0.8; 0.43>**	**<0.9; 0.44>**
% Deleted Values	1.05	1.05	0.98	0.99
CPU time	**0.52**	**0.51**	0.61	0.92

Therefore, we should note that for our approach, even though all the constraints are called upon, the local decisions are quickly propagated globally to prove the inconsistency.

We will try to carry out other experiments of our approach, on any kind of constraints (predicate, positive constraints, negative constraints, etc.) to check its efficiency.

6 Conclusion

Even though the major importance of arc-consistency approaches for improving the search of solutions in many NP-Complete problems and the advents of both distributed computing and networking technologies, there does not exist any distributed approach that efficiently achieve arc-consistency on general constraint networks.

Thus, in this paper we propose a new approach based on DRAC for general CNs. The aforementioned experimental comparative evaluation shows the high performance of GDRAC as a distributed arc-consistency technique, which removes all the arc-inconsistent values while inquiring least time and constraint checks for constraints expressed in extension. We abode well for our approach to efficiently handle any of the other forms of constraints: by a conjunctive constraint, by an arithmetic relation or by a predicate for which no particular semantics in known (data base query, user's context-dependent constraint, etc.). The perspectives of this work are to propose an improvement of this schema to be able to efficiently handle higher levels of consistency.

References

1. Briot, J.P. Actalk: A Framework for Object-Oriented Concurrent Programming – Design and Experience. Proceedings 2nd France-Japan Workshop (OBPDC'97)", 23 pages, 1999.
2. Bessière, C., Freuder, E.C., Régin, J-C. Using inference to reduce arc consistency computation. Proceedings IJCAI-95, pages 592-598, 1995.
3. Bessière, C., Freuder, E., Régin, J-C. Using constraint Metaknowledge to Reduce Arc Consistency Computation. Artificial Intelligence, Vol. 107, p125-148, 1999.
4. Bessière, C., Régin, J-C. Arc consistency for general constraint networks : preliminary results. Proceedings IJCAI'97, pages 398-404, 1997.
5. BenHassine, A., Ghedira, K. How to Establish Arc-Consistency by Reactive Agents. Proceedings of the 15th ECAI-02, p. 156-160, 2002.
6. BenHassine, A., Ghédira, K. Using Reactive Agents to Establish Arc-Consistency. Proceedings of the 7th PRICAI-02, p.97-107, 2002.
7. Dechter, R., Pearl, J. Tree-Clustering Schemes for Constraint-Processing. Proceedings AAAI 1988.
8. Lamport, L. and Chandy, K. M. Distributed snapshots: Determining global states of distributed systems. TOCS, 3(1): 63-75, Feb 1985.
9. Mackworth, A. K. Consistency in networks of relations. Artificial Intelligence, 8, 99-118, 1977.
10. Mackworth, A. K. On reading sketch maps. Proceedings IJCAI'77, pages 598-606, 1997.
11. Montanari, U. NetWorks of Constraints: Fundamental properties and applications to picture processing. Information Sciences, 7, P.95-132, 1974.
12. Mohr, R., Henderson, T. C. Arc and path consistency revisited. Artificial Intelligence, 28, p225-233, 1986.
13. Mohr, R., Masini, G. Good old discrete relaxation. Proceedings ECAI'88, pages 651-656, 1988.
14. Regin, J. C. A filtering algorithm for constraints of difference in CSPs. Proceedings AAAI'94, pages 362-367, 1994.
15. Van Beek, P., Dechter, R. On the minimality and global consistency of row-convex constraint networks. Journal of the ACM, 42(3):543-561, 1995.

A Systematic Search Strategy for Product Configuration

Helen Xie[*], Philip Henderson, Joseph Neelamkavil, and Jingxin Li

Integrated Manufacturing Technologies Institute
National Research Council of Canada
800 Collip Circle, London, Ontario, Canada N6G 4X8

Abstract. Constraint satisfaction problem (CSP) paradigm has proven highly successful in product configuration, particularly for build-to-order products, by assigning component types to all components without violating any constraints. For engineer-to-order products, however, product configuration requires assigning design parameters to each component as well. Hence, it often involves numeric variables, n-ary constraints, and constraints over variables that depend on other variables. Thus, an efficient search strategy is needed to address these issues. In this paper, an extension to the CSP, called Dependent CSP, is proposed to accommodate the complex engineer-to-order product configuration and the search strategy. In the Dependent CSP, variables are categorized as independent variables and dependent variables so that, search space can be reduced by eliminating dependent variables. Backjumping search strategy is employed to search for a solution as effective as possible. An updating mechanism is designed to avoid repetitive and unnecessary variable updating and constraint evaluation. Several variable ordering heuristics are assessed and the most effective ones are chosen for solution implementation. By applying these strategies, we can achieve a very efficient search algorithm for product configuration. The algorithm has been applied in a product configuration problem – an elevator system design – and a configuration solution can be obtained in a matter of seconds.

Keywords: Constraint satisfaction, product configuration, numeric variables, n-ary constraints, dependent variables, backjumping, variable ordering

1 Introduction

The product configuration is an enabling technology in achieving mass customization to meet the challenges of global competition and customer satisfaction. It is intended to aid manufacturing companies to configure customized products quickly and efficiently by automating the configuration process as much as possible. The constraint satisfaction problem (CSP) paradigm has proven highly successful for product configuration. It provides several advantages [11] in terms of problem representation, algorithms, and result evaluation. First, the domain knowledge of product configuration can be represented in a declarative form, which makes the configuration problem easy to define and maintain. Second, the search strategies are,

[*] Corresponding author. *E-mail address: helen.xie@nrc.gc.ca*

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 718-727, 2004.

generic and domain independent. By experimenting with different search strategies efficient search algorithms can be identified for a certain type of product configuration problem. Third, given an existing configuration, its accuracy can easily be verified by checking the consistency of constraints. Finally, CSP algorithms easily allow the generation of multiple alternative solutions for preference and optimization purposes.

A typical constraint satisfaction problem (CSP) is defined by a set of variables, $X = \{x_1,\ldots, x_n\}$, and a set of constraints, C, over these variables. An associated domain, D_i, contains possible values for x_i. A constraint $c\ (x_{i,},\ldots, x_j) \in C$ specifies a subset of the Cartesian product $D_i \times \ldots \times D_j$ indicating the variable assignments that are compatible with each other. A *solution* to a CSP is a complete *assignment* of values to the variables such that all constraints are simultaneously satisfied [7]. In a product configuration framework, component types are represented as variables with discrete and finite domains, and compatibilities of various components to form a valid configuration are represented as constraints [9]. For example, an elevator door is available in four models: SSSO, 2SSO, SSCO, and 2SCO, and an elevator platform can be chosen from three models: 2.5B, 4B, and 6B. The constraints for compatible models are represented as tuples: {SSSO, 2.5B}, {2SSO, 4B}, {SSCO, 4B}, {2SCO, 6B}. A valid configuration is constructed by choosing a door model and a platform model such that the combination appears in one of the tuples.

For engineer-to-order products, however, product configuration also requires assigning design parameters for each component. While the design parameters, including but not limited to component types and component quantities, are represented as variables, design constraints restricting design parameter assignments can be represented as constraints. Since constraints over design parameters may take a wide variety of formats, modeling and solving constraints for engineer-to-order products presents several challenges:

1. Variables may be defined with continuous and numeric domains.
2. Constraints may be n-ary, meaning more than two variables may appear in a constraint.
3. Constraints may be represented as mathematical expressions or computable procedures.
4. A constraint may be defined over variables whose existence depends on the values chosen for other variables. This type of constraint is called activity constraint [8] or conditional constraint [5].
5. More generally, a constraint may be defined over variables whose values cannot be independently assigned, because those variables have dependent relations with other variables outside the constraint. This type of constraint is called dependent constraint.

To address the issue of the conditional constraint, several constraint models have been proposed to extend the original configuration framework, including Dynamic CSP [8], Generative CSP [4][12], Composite CSP [10], and Mixed and Conditional CSP [5]. While the Dynamic CSP, Generative CSP, and Composite CSP models formulate the conditional constraint in such ways that effective search algorithms can be employed, the Conditional CSP model reduce the conditional constraints to a set of standard CSPs by analyzing dependencies between the conditional constraints and applying the conditional constraints in order.

The issues on n-ary constraints over numeric variables or mixed variables (both discrete and numeric variables) are also addressed in Mixed and Conditional CSP [5]. In order to achieve arc-consistency, the n-ary constraint can be decomposed into an equivalent network of ternary constraints. When all but two variables of the n-ary constraint have been instantiated, a binary refine operator can be applied.

In an n-ary dependent constraint, some variables (called dependent variables) may not be independently assigned values from their domains, as they depend on other variables through dependent relations. The dependent relations are often represented by mathematical expressions or computable procedures. In the dependent relations, some variables can be independently assigned values from their domains. Although these variables may not have appeared in the dependent constraint, they are considered as independent variables in the dependent constraint, since these variables have the potential to make the dependent constraint satisfied. In the dependent constraint, a constraint check cannot be done immediately when an independent variable is assigned a value, as its associated dependent variables have to be updated as well. Unlike variables of the conditional constraint which have only two values (existence or non-existence) to choose from, the dependent variables may have numeric domains. Converting the n-ary dependent constraint with numeric variables into a set of typical constraints with tuples may result in exponential increase in parameters and constraints [1]. Hence, a generic constraint model and efficient search strategy are needed for the n-ary dependent constraint.

The main contribution of this paper is to propose a generic constraint model and an efficient search strategy for product configuration with n-ary dependent constraints over numeric variables. Since a dependent constraint can be ultimately represented by a set of independent variables through dependent relations, and dependent variable cannot be independently assigned a value, the search space could be reduced by eliminating dependent variables. As dependent relations among variables are represented by mathematical expressions and computable procedures, their formats only become known later at product constraint modeling time. In order to make a search algorithm as generic as possible, a systematic search strategy is employed. Specifically, backjumping algorithm is chosen because of its success in avoiding thrashing (repeated failure due to the same reason) which is often the leading factor in search efficiency. During a search process, a consistency check is required, whenever an independent variable is assigned a new value. Since a constraint is linked to independent variables through dependent variables, the dependent variables are frequently updated. Thus, the updating of dependent variables becomes a bottleneck in search efficiency. An efficient updating mechanism is designed to avoid unnecessary updating. Moreover, several variable ordering heuristics were assessed and implemented. By applying these strategies, we can achieve a very efficient search algorithm for product configuration. The algorithm has been applied in a product configuration problem – an elevator system design, with excellent results: the search can be done in a matter of seconds.

The remainder of the paper is organized as follows. Section 2 defines the constraint model applicable to n-ary dependent constraints. In Section 3, we present a backjumping search algorithm for product configuration, provide a mechanism for updating dependent variables and constraints, and discuss variable ordering heuristics. A case study for configuring an elevator system is presented in section 4. Finally, conclusions are given in section 5.

2 A Constraint Model for Product Configuration

As a search algorithm should be generic to any product configuration, a constraint model is necessary for defining product configuration. Here, product configuration can be represented by a Dependent Constraint Satisfaction Problem as follows:

Definition. A Dependent Constraint Satisfaction Problem is defined as <X, D, R, C>, where

- $X = \{x_1, x_2, \ldots, x_n\}$ is a finite set of variables,
- Each $x_i \in X_{in}$ can take its value from a finite domain D_i, where $D_i \in D$, $X_{in} \subseteq X$ is a set of independent variables,
- Each variable $x_j \in X_{de}$ depends on its dependent relation $r_j \in R$ to its ancestors $X_a \subseteq X$, $X_{de} \subseteq X$ is a set of dependent variables. $X=(X_{in} \cup X_{de})$, and $(X_{in} \cap X_{de})=0$.
- A set of constraints C restricts the combination of values that variables can take.
- A solution to a Dependent CSP is an assignment of a value from its domain to every variable from X, in such a way that every constraint from C and every dependent relation from R are satisfied.

Variables are modifiable during a search process to satisfy all the constraints, so that a solution can be found. According to the way variables can be modified, they can be classified as an independent variable (IV) or a dependent variable (DV). The independent variable is a variable that may be directly modified by a search algorithm. Its value can independently be assigned within its domain. The dependent variable depends on a dependent relation. It can only be derived from existing variables. Its value ultimately depends on independent variables and cannot be independently modified. An ancestor of a dependent variable is a variable whose value determines (at least in part) the value of the dependent variable. Direct ancestors can be any combination of IVs and DVs, but ultimately a dependent variable is defined by IVs. There are advantages to separating dependent variables from independent variables. First, only independent variables define the search space, so the number of possible combinations is dramatically reduced. Second, when a dependent variable is used by multiple constraints, it needs to be computed only once. These characteristics help to improve the efficiency of the search algorithm.

Variables can have numeric or non-numeric domains. Examples of numeric variables are choices relating to dimension or weight, while non-numeric variables can be the model of a part. Numeric variables are given a range of possible numeric values specified by a minimum, maximum, and an interval to be used from one value to the next, while non-numeric variables each have a list of possible values. The sequence of the values in their domain will determine the order in which values are tried in a search process.

Dependent relations are represented by mathematical expression or computable procedures, such as formulas, tables, etc. They specify design relations among independent variables and dependent variables or among dependent variables. An example of a dependent relation is shown as follows: *Counterweight Plate Weight = 0.2816T(D(BG-2)-3.5(D-5)-6(D-7))*, where *BG* is the *Distance Counterweight Between Guiderails* (an independent variable), *D* is the *Counterweight Plate Depth* (an independent variable), *T* is the *Counterweight Plate Thickness* (a constant), *and Counterweight Plate Weight* is a dependent variable .

Constraints specify the restrictions that must be satisfied for a solution. The restrictions may represent a logical requirement, physical requirement, compatibility among parts, safety regulations, or any other design requirement that may be required. A constraint may be extensionally represented as tuples, or intensionally described by mathematical expressions or computable procedures that indicate a valid or invalid assignment for consistency check. The difference between constraints and dependent relations is that constraints specify a limit, while dependent relations result in a value. For example, a constraint can be stated as follows: the *Platform Width* must be at least 60 inches. A constraint may apply to any number of variables, including any combination of IVs and DVs. However, since dependent variables are deterministically defined by independent variables, the constraint ultimately depends solely on independent variables. The number of independent variables that affect a constraint is called the constraint's arity. Sometimes, a independent variable may not appear in a constraint explicitly, since it may affect the constraint through dependent variables. Nevertheless, the relevant independent variables can still be identified by searching the ancestors of a constraint's dependent variables. The relationship between constraints and independent variables is many-to-many, meaning that a constraint may depend on multiple independent variables, and an independent variable may affect multiple constraints.

3 A Search Strategy

Once product configuration has been formulated as a constraint satisfaction problem, a solution can be found using search algorithms. Since they are represented by mathematical expressions or computable procedures, dependent constraints and their corresponding dependent relations can take a wide variety of formats. Hence, they are not known during algorithm design time. To provide a generic search algorithm for solving product configuration problems, we use systematic search strategies for product configuration.

A systematic search strategy incrementally extends a partial solution towards a complete solution by repeatedly choosing a value for another variable, consistent with the values in the current partial solution [2]. Since it traverses the search space systematically, the advantage is that a solution, if one exists, can eventually be found. Also, the algorithm is general and applicable to any configuration design problems. As previously described, dependent variables can be eliminated from search space. Thus, only independent variables are considered as variables in search algorithms.

Backtracking is a primary algorithm in systematic search. It has two phases: a forward phase in which the next variable is selected and the current partial solution is extended by assigning a consistent value, if one exists for the next variable; and a backward phase in which, when no consistent solution exists for the current variable, attention returns to the previous variable assigned [3]. Backtracking suffers the drawback of thrashing, i.e. repeated failure due to the same reason. The efficiency of backtracking algorithm was improved by backjumping, a proper updating mechanism, and variable ordering in configuration design.

3.1 Backjumping

Backjumping improves on backtracking by analyzing the reasons for a dead-end and jumping back to the appropriate variable. In backtracking, a dead-end is encountered when a consistent value cannot be found for the next variable (i.e. the current partial solution cannot be extended). Instead of just going back to the preceding variable in the ordering, the backjumping algorithm tries to identify the source of failure and prunes a large portion of search space without missing any potential solutions. To help determine an appropriate backtrack point, we discuss the following three situations:

1. *A dead-end variable breaks one unary constraint.* The unary constraint cannot be affected by other variables, so if this is the case, this constraint shall never be satisfied and the CSP is impossible to solve.
2. *A dead-end variable breaks one constraint with n-ary variables.* The broken constraint has one or more variables that can affect it (excluding the dead-end variable). Although any of these variables could be modified, the algorithm should not skip any possible solutions. Thus, the algorithm should jump back to the closest previous variable for this constraint. If the algorithm moves farther back, it may skip a potential solution, and any jumps that do not go back beyond this point will be futile since this constraint will fail again.
3. *A dead-end variable breaks more than one constraint with n-ary variables.* In this case, the values of the variable can not become valid unless all broken constraints are affected. Thus, jumping back to the closest previous variable among all broken constraints is not adequate, since it does not affect the constraints whose variables appear before that variable. In order to ensure that all constraints are affected, the algorithm should jump back to the farthest variable, called the cutoff variable, among the closest variables of all broken constraints. The algorithm cannot jump back farther without facing the risk of missing potential solutions. After jumping back, still, if none of the values are compatible with at least one constraint for the cutoff variable, then the algorithm should jump back to the closest variable among any connected constraints for the new current/dead-end variable.

3.2 Updating Mechanism for Consistency Checks

While choosing appropriate backtrack points could potentially prune a large portion of search space, determining the timing for constraint consistency checks can also improve the efficiency of the search algorithm. It is necessary to have an updating mechanism that identifies which constraints and dependent variables have been affected by the change of an independent variable's value, and updates their status accordingly. The efficiency of the updating mechanism has a major impact on the overall efficiency of the search algorithm, since updating is performed frequently (every time an IV's value is changed).

However, enabling each dependent variable of a constraint to re-compute itself does not guarantee the values will be properly updated, as the constraint and dependent variables need to be updated after their ancestors. In our previous approach [13], a list of dependent variables is stored for each variable (either independent variable or dependent variable) as its direct descendants. Whenever an independent variable is modified, it calls the update procedures of its direct descendants, which in turn call the update procedures of their direct descendants, and so on. In this way, every dependent variable will be properly updated and correctness of the constraint's status is guaranteed. However, this approach may still update a dependent variable more than once. For example, suppose we have (A -> B, C) and (C -> B). An arrow indicates dependency: (A -> B) means that B is a direct descendant of A (or, equivalently, A is a direct ancestor of B). Once A is updated properly, and B's update procedure is called and followed by C's update procedure since B and C are A's direct descendants. However, B is C's direct descendant as well, so B's update procedure shall be called once more, right after C's update procedure. Consequently, B's update procedure was called more than once, because A does not know which of its direct descendants to be updated first.

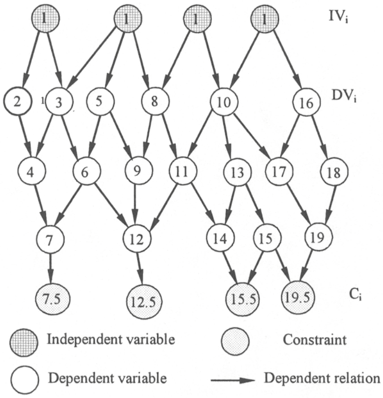


Fig. 1. A directed acyclic graph of independent variables, dependent variables and constraints

In our current approach, the dependencies among variables and constraints are considered as arcs in a directed graph, where variables and constraints are nodes and there is an arc from every variable to each one of its descendants (Fig. 1). In the directed graph, independent variables do not have any ancestors, and constraints do not have any descendants. Dependent variables can only be derived from independent variables or other dependent variables that in turn are eventually derived from independent variables. This directed graph can be shown to be acyclic, implying that a topological ordering exists. A topological ordering of the directed acyclic graph provides an updating order which guarantees that each variable and constraint's status are correct and need to be computed only once. Multiple topological orderings are valid, but the recommended ordering is formed by minimizing the value given to constraints and dependent variables, so that constraints can be evaluated as early as possible.

As previously described, constraints are often indirectly linked to independent variables through dependent variables in configuration design. Also a dependent variable may depend on one or more independent variables. Thus, if any of the independent variable ancestors for a dependent variable is not instantiated, the dependent variable cannot be used as an authentic source for evaluating associated constraints. The evaluation of a constraint has to wait until the last independent variable ancestor is instantiated. Using the criteria for determining proper timing for consistency checks, unnecessary repeated updating can be avoided.

3.3 Variable Ordering

The performance of the backjumping algorithm can also be improved by choosing the order of variable instantiation [6]. In the algorithm, variable ordering is used as a preprocessing technique. A fixed order is determined by heuristic approaches prior to starting of the search. Several heuristics have been analyzed for selecting variable order. One consideration is the variable's degree, a number of variables that are connected with it. The maximum degree variable is instantiated first. If variables are tied in the first heuristic, then a variable with the fewest domain values would be chosen as a secondary heuristic. However, the success of these heuristics is not independent of the specific product configuration problem; hence, the search algorithm may have to try several orderings before finding a good variable ordering.

4 A Case Study and Experimental Results

To exam the efficiency of the constraint model and search algorithm described above, we have tested a configuration design problem—configuring elevator systems [14]. The configuration process begins with a list of customer requirements, such as elevator car capacity and speed, and building dimensions. To configure an elevator system, one must assign a set of variables that satisfies both customer requirements and design constraints. In product configuration problems, not all variables are compatible, and certain combinations may not meet functional or safety regulations. The algorithm has to modify variables until it achieves a valid configuration.

In order for the search algorithm to find a valid solution for the elevator design, it is necessary to generate associated product definitions in the constraint model. There are 241 variables in the elevator system. Among these, there are 32 independent variables (such as platform model and counterweight buffer quantity), and 184 dependent variables (such as counterweight quantity and hoist cable quantity), and 25 input variables (such as car capacity and car speed). Input variables capture customer requirements and are considered as fixed values upon entering the system. There are also 50 constraints that establish criteria for functional and safety regulations, which guide the search algorithm to find a valid solution. In addition to variables and constraints, there are also dependent relations, such as mathematical expression or tables, between dependent variables and independent variables. These relations define how the dependent variables are derived from independent variables.

The backjumping search algorithm was implemented in Java using IBM VisualAge for Java 4.0. It solved almost all elevator configuration problems that we tried in less than 15 seconds. The only problematic case is when car capacity and car speed inputs are set to their maximum possible values. For this scenario, the best variable ordering (we could find) took 50 seconds, whereas the automated variable ordering never completed the search. A series of tests was performed on an Intel Pentium 4 CPU, 1.8GHz, and 1G RAM running on Windows 2000. The results below show that the algorithm works quite well with an automated variable ordering. Note that these results are the slowest test cases found for the given car capacity and car speed. For instance, other test cases with car capacity at 4000lbs and car speed set to 400 feet per minute found solutions in 10-15 seconds.

Table 1. Backjumping results (with automated variable ordering)

Worst-case time found (seconds)		Car Speed (feet per minute)				
		200	250	300	350	400
Car Capacity (pounds)	2000	1.5	----	----	----	3.4
	3000	----	1.5	----	4.5	6.2
	4000	2.4	----	2.5	----	Forever

A web-based application prototype system was implemented using this algorithm. The system allows the customer to enter requirements, and displays the final configuration results back to the customer through the Web. The Web application was deployed on IBM WebSphere Application Server.

5 Conclusions

Market trends that affect today's competitive environment are changing dramatically. Mass production of identical products - the business model for industries in the past - is no longer viable for many sectors. Customized products offer great market potential to manufacturers in the current climate of global competition and improved customer satisfaction. The complexity of products brings along new demands for configuration technology to cope with search efficiency. However, commercially available configuration systems only support build-to-order type product configuration in which constraints are represented by tuples. In this paper, an extension of the CSP paradigm was presented to cover dependent constraints with mathematical expressions in product configuration. The extension supports n-ary dependent constraints and variables with both discrete and numeric domains. Dependent variables are separated from independent variables to reduce search space. The updating mechanism proposed for dependent variables and constraints ensures correctness while avoiding repeated computations. The search algorithm is based on backjumping, a systematic search strategy. Specific backjumping situations were discussed to cover many-to-many relations between variables and constraints. Several heuristics of variable ordering were also applied for the backjumping search algorithm. The implemented algorithm is capable of solving almost all elevator configuration problems within 15 seconds based on an elevator case study. The test results show that the algorithm works well with a good (automated) variable ordering heuristic. This approach can be easily applied to a wide variety of product configuration problems.

References

1. Bartak, R.: Theory and Practice of Constraint Propagation. Proceedings of CPDC2001 Workshop (invited talk). Gliwice (2001) 7-14
2. Bartak, R.: Constraint Programming: In Pursuit of the Holy Grail. Proceedings of Week of Doctoral Students (WDS99), Part IV. MatFyzPress, Prague (1999) 555-564
3. Dechter, R. and Frost, D.: Backjump-based Backtracking for Constraint Satisfaction Problems. Artificial Intelligence, Vol. 136 (2002) 147-188
4. Fleischanderl, G., Friedrich, G., Haselboeck, A., Schreiner, H. and Stumptner, M.: Configuring Large Systems Using Generative Constraint Satisfaction. IEEE Intelligent Systems, Vol. 13 (1998) 59-68
5. Gelle, E. and Faltings, B.: Solving Mixed and Conditional Constraint Satisfaction Problems. Constraints, Vol. 8, No. 2 (2003) 107-141
6. Kumar, V.: Algorithms for Constraint Satisfaction Problems: A Survey. AI Magazine, Vol. 13, No. 1 (1992) 32-44
7. Miguel, I. and Shen, Q.: Solution Techniques for Constraint Satisfaction Problems: Foundations. Artificial Intelligence Review, Vol. 15 (2001) 243-267
8. Mittal, S. and Falkenhainer, B.: Dynamic Constraint Satisfaction Problems. Proceedings of the 8th National Conference on Artificial Intelligence (1990) 25-32
9. Mittal, S. and Frayman, F.: Towards a Generic Model of Configuration Tasks. Proceedings of the 11th IJCAI, Detroit, MI (1989) 1395-1401
10. Sabin, D. and Freuder, E. C.: Configurations as Composite Constraint Satisfaction. Working Notes, AAAI Fall Symposium on Configuration, Boston (1996) 28-36
11. Stumptner, M.: An Overview of Knowledge-Based Configuration. AI Communications: The European Journal on Artificial Intelligence, Vol. 10, No. 2 (1997) 111-125
12. Stumptner, M. and Haselbock, A.: A Generative Constraint Formalism for Configuration Problems. 3rd Congress Italian Assoc. for AI. Torino, Italy. Lecture Notes in AI, Vol. 729. Springer-Verlag (1993) 302-313
13. Xie, H. and Lau, F.: Towards Engineer-to-order Product Configuration. Proceedings of the ISCA 15th International Conference, Computer Application in Industry and Engineering. San Diego, CA, USA (2002) 180-184
14. Yost, G. R. and Rothenfluh, T. R.: Configuring Elevator Systems. Int. J. Human-Computer Studies, Vol. 44 (1996) 521-568

A Bayesian Framework for Groundwater Quality Assessment

Khalil Shihab[1] and Nida Al-Chalabi[2]

[1] School of Computer Science & Mathematics, Victoria University, Australia,
kshihab@scm.vu.edu.au
[2] Department of Computer Science, SQU, Oman,
nida@squ.edu.om

Abstract. There has been an increasing interest in the monitoring and the assessment of surface and groundwater quality. Experts in this area have been arguing that the current used techniques are not accurate means of measuring water contamination. This is mainly because these techniques neglect the characteristics that are significant in understanding of pollution-generation processes, which is stochastic in nature, from various sources. In particular, these techniques emphasize neither the stochastic nature of the water contamination process nor the precision and the accuracy of the tested methods used by environmental laboratories. In this work, we describe the development and the application of a prototype Bayesian Belief Network (BBN) that models groundwater quality in order to assess and predict the impact of pollutants on the water column. The methods presented are widely applicable and handle many of the problems encountered with other methods.

1 Introduction

Declining surface and groundwater quality is regarded as the most serious and persistent issue affecting Oman in particular. The Sultanate faces severe challenges as it confronts the extremely growing and complicated issues of contamination of the groundwater supply in and around hazardous waste disposal sites across the nation. There are many observable factors, which contribute to deteriorating water quality, that need to be monitored and their maximum allowable limits need to be determined. Decline in water quality is manifested in a number of ways, for example, elevated nutrient levels, acid from mines, domestic and oil spill, wastes from distilleries and factories, and temperature. These factors and others will provide the input data for our computer system.

Groundwater quality and pollution are determined and measured by comparing physical, chemical, biological, microbiological, and radiological quantities and parameters to a set of standards and criteria. A criterion is basically a scientific quantity upon which a judgment can be based [1].

Many attempts have been made by various states to develop satisfactory procedures for assessing, monitoring and controlling contamination of the groundwater supply in and around hazardous waste disposal sites [2]. These attempts

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 728-738, 2004.

resulted in various environmental regulations that focus attention on the maximum allowable limits of hazardous pollutants in the groundwater supply. While on the other hand, they pay scant attention to the nature of groundwater data and the development of valid statistical procedures for detecting and monitoring groundwater contamination.

Recent attempts based on Artificial Intelligence (AI) were first applied to the interpretation of biomonitoring data [3]. Other works were based on pattern recognition using artificial neural networks (NNs). More recent study described a prototype Bayesian belief network for the diagnosis of acidification in Welsh rivers [3].

Bayesian methods of statistical inference offer the greatest potential for groundwater monitoring. This is because these methods can be used to recognize the variability arising from three different sources of errors, namely, analytical test errors, sampling errors and time errors, in addition to the variability in the true concentration [4]. The Bayesian methods can also be used to significantly increase the precision and the accuracy of the test methods used in a given environmental laboratory [5]. Furthermore, these methods are simple to apply and have sufficient flexibility to allow reaction to scientific complexity free from impediment from purely technical limitations.

2 Data Collection and Pre-processing

2.1 Data Collection

The Oman Mining Company (OMCO), Ministry of Environmental and Regional Municipalities (MRME) and the Department of Earth Science, Sultan Qaboos University, maintain data on the concentration of the harmful substances in the groundwater at Taqah monitoring sites, which is allocated to the south of the sultanate of Oman. We observed that good quality data were obtained from several monitoring wells in this region. Because of the lack of monitoring wells in that area, we filled in the missing measurements with data obtained from Oman Mining Company (OMCO) and Ministry of Environmental and Regional Municipalities (MRME).

The data were collected from these monitoring wells in the Sultanate identified to be important in assessing the groundwater quality and in the prediction of the effect of certain pollutants on drinking water. The period covered in these locations is from 1994 to 2002. Each site has several monitoring wells and water samples were collected periodically from these wells and the concentration of the pollutants in these water samples was recorded.

2.2 Data Pre-processing Using Bayesian Reasoning

Data for water quality assessment are normally collected from various monitoring wells and then analyzed in environmental laboratories in order to measure the concentration of a number of water quality constituents. We realized that the methods used by these laboratories do not emphasize the accuracy. Therefore, we used a modified Bayesian model to that was developed by Banerjee, Plantinga and Ramirez [10], for preprocessing the data that used for the development of the Bayesian Belief Network.

2.2.1 Bayesian Models

The model is as follows:

Let S denote a particular hazardous constituent of interest. Since the concentration of the substance may vary from well to another, it is necessary to consider each well separately. Let $\aleph_t=(\aleph_{t1}, \aleph_{t2}, \aleph_{t3}, ..., \aleph_{tm})$ be the vector of m measurements of the concentration of S in m distinct water samples from a given well at a given sampling occasion where (m>=1) and (t=1,2,...). Each measurement consists of the true concentration of S plus an error.

Let X_t be the true concentration of S in the groundwater at sampling occasion t.

Taking the assumption that the true concentration X_t is unknown and is a random variable, the model evaluates the posterior distribution of X_t given the sample measurements $\aleph_t$ at sampling occasion t. All published work in the context of groundwater quality data rested on the normality assumption. That is, given $X_t = \aleph_t$ and δ^2, the concentration measurements in $\aleph_t$ represent a random sample of size m for random distribution with mean $\aleph_t$ and variance δ^2.

Since the concentration of the substance S in water samples obtained at different sampling occasions might vary considerably, we assume that the parameters $\aleph_t$ and δ^2 of the normal distribution are random variables with certain prior probability distribution. To model these prior distributions, we also used the natural conjugate families of distribution for sampling from a normal distribution. Therefore, the model for prior distribution of X_t and δ^2 can be presented as follows:

For t =1 ,2 ,… and given δ^2 the conditional distribution of X_t at sampling occasion t is a normal distribution with mean μ_{t-1} and variance $\delta^2_{t-1}\,\delta^2$, and marginal distribution of δ^2 is an inverted gamma distribution with parameter β_{t-1} and v_{t-1}.

This model uses the following prior distribution, which represents the concentration measurements before the first sampling.

The pdf of the prior distribution of X_0 is:

$$f_0(x_0) = \left\{1+\frac{1}{2v_0}\left[\frac{x_0-\mu_0}{\sigma_0\sqrt{\beta_0/v_0}}\right]^2\right\}^{-(2v_0+1)/2} \qquad (2.1)$$

,

which is the pdf of the student's t-distribution with $2v_0$ degrees of freedom, location parameters μ_0 and variance $\delta_0^2\beta_0/v_0$.

Now suppose that the observations are available on the concentration of S, given the sample X_t the posterior marginal distribution of X_t is a student's t-distribution with $2v_t$ degree of freedom, location parameters μ_t and variance $\delta_t\ \beta_t/v_t$ where the pdf has the form:

$$f_t(x_t/x) = \left\{1 + \frac{1}{2v_t}\left[\frac{x_t - \mu_t}{\sigma_t\sqrt{\beta_t/v_t}}\right]^2\right\}^{-(2v_t+1)/2} \tag{2.2}$$

where:

$$\begin{aligned}
\beta_t &= \beta_{t-1} + \sum_{j=1}^{m}(x_{tj} - \bar{x})/2 + m(\mu_{t-1} - \bar{x}_t)/\left[2(1+m\sigma_{t-1}^2)\right] \\
v_t &= v_{t-1} + m/2 \\
\mu_t &= (\mu_{t-1} + m\bar{x}_t\sigma_{t-1}^2)/(1+m\sigma_{t-1}^2) \\
\sigma_t^2 &= \sigma_{t-1}^2/(1+m\sigma_{t-1}^2) \\
\bar{x}_t &= \sum_{j=1}^{m} x_{tj}/m
\end{aligned} \tag{2.3}$$

We can see obviously from the equation of μ_t the sequential nature of this posterior distribution. That is, at each sampling occasion t, when more new information about concentration of S in the groundwater becomes available.

The posterior distribution is revised forming a recursion process. This process of updating the posterior distribution may be continued indefinitely when new data x_t becomes available

To convince the others about the true unknown concentration of the substance S in the well under consideration, it is frequently more convenient to put a range (or interval) which contains most of the posterior probability. Such intervals are called highest posterior density (HPD) intervals. Thus for a given probability content of (1- α) ,0< α<1, a 100(1- α) percent HPD interval for X_t, is given by :

$$\mu_t \pm t_{2v_t}(\alpha/2)\sigma_t\sqrt{\beta_t/v_t} \tag{2.4}$$

when $t_{2vt}(\alpha/2)$ is the 100(1- α/2) percentile of the student's t-distribution with 2vt degree of freedom.

2.2.2 The Bayesian Algorithm

In brief, the monitoring algorithm, which is based on the Bayesian model, is as follows:

1. Fix a value of α (0< α <1) based on the desired confidence level. In this case, we chose α to be 0.01 .
2. Since we do not have enough data to work with, we used the same parameters of the prior distribution used in the model of Banerjee, Plantinga and Ramirez. These parameters are :

   β_0= 0.0073 , v_0=2.336 , μ_0= 9.53 , δ_0^2 =3056.34 .

3. At each sampling occasion t , (t= 1,2,...) , compute the parameters β_t , ν_t , μ_t and δ_t of the posterior distribution X_t given the set of observations in $\aleph_t$ on the concentration of S available from a given well in a given site using (2.3). Compute LHPD and UHPD using these parameter estimates and (2.4).
4. Plot μ_t, LHPD, and UHPD that are obtained in step 3 above against sampling occasion t.
5. For the next sampling occasion, update the values of the parameters β_t, ν_t, μ_t and δ_t using (2.3) and the set of data just obtained. Then recomputed LHPD, and UHPD using the updated parameter values in (2.4) and repeat step 4 above.

We have applied this algorithm on the data that were collected from Salalah in the Sultanate of Oman. It is expected that the data from each well is not normal, but each one is taken from a normal distribution. Some of these data needed to be scaled down to simplify the process and to have a smooth graph so that we can study them easily. For this purpose, we have used the following normalization technique:

$$x = \frac{\bar{x} - \mu}{\sigma}, \text{ where } \bar{x} = \sum_{i}^{n} x_i \Big/ n \text{ and } \sigma = \sqrt{\frac{\sum_{i}^{n} x_i^2 - n\bar{x}^2}{n-1}}$$

2.2.3 Algorithm Implementation

The pre-processing system is implemented on PC platform using Visual Basic programming language.

Table 1 presents the concentration data for TDS (Total Dissolved Solids) in Salalah for Well 001/577. In particular, the table shows the true concentration data for TDS that was produced by our pre-processing system.

3 Bayesian Belief Networks (BBNs)

After the pre-processing stage, we constructed and used a Bayesian Belief Network (BBN) in order to predict the impact of pollution on groundwater quality.

Bayesian Belief Networks are effective and practical representations of knowledge for reasoning under uncertainty. There are a number of successful applications of these networks in such domains as diagnosis, prediction, planning, learning, vision, and natural language understanding [6].

Bayesian Belief Networks (see Figure 1) are graphical structures used for representing expert knowledge, drawing conclusions from input data, and explaining the reasoning process to user. These networks are also called knowledge maps, probabilistic causal networks, and qualitative probabilistic networks [7]. They have been an increasingly popular knowledge representation for reasoning under uncertainty. A BBN is a directed acyclic graph (DAC) whose structure corresponds to the dependency relations of the set of variables represented in the network (nodes).

Each node in a belief network represents a random variable, or uncertain quality, that can take two or more possible values. The arcs signify the existence of direct influences between the linked variables, and the strengths of these influences are quantified by conditional probabilities. These links can be said to have a causal meaning.

Table 1. Concentration Data for TDS (Total Dissolved Solids) in Salalah, Well 001/577

Date	Observed Concentration	LHPD	Expected True Concentration	UHPD
May-86	1.147	0.85	1.15	1.45
Oct-86	1.106	1.00	1.13	1.26
Apr-88	1.938	1.12	1.40	1.68
Oct-88	2.237	1.33	1.61	1.88
Apr-90	3.857	1.60	2.06	2.52
Oct-90	3.834	1.91	2.35	2.79
Apr-92	3.957	2.18	2.58	2.98
Oct-92	3.761	2.38	2.73	3.08
Apr-94	4.3	2.58	2.90	3.23
Oct-94	3.958	2.72	3.01	3.30
Apr-96	1	2.54	2.83	3.11
Oct-96	3.714	2.64	2.90	3.16
Apr-98	3.65	2.73	2.96	3.19
Nov-99	3.381	2.78	2.99	3.20
Dec-99	3.396	2.83	3.02	3.20
Nov-00	3.477	2.87	3.04	3.22
Dec-00	3.498	2.91	3.07	3.23
Nov-01	3.23	2.93	3.08	3.23
Dec-01	3.243	2.95	3.09	3.22
Jan-02	3.267	2.97	3.10	3.22
Feb-02	3.297	2.99	3.11	3.22

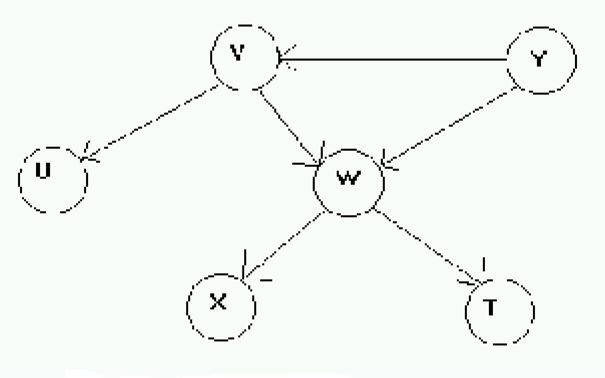


Fig. 1. A simple BBN

The graph in Figure 1 represents the following joint probability distributions of the variables V, Y, U, W, X and T.

$$P (U, V, Y, W, X, T) = P (T/W). P (X/W). P (W/V, Y). P (U/V). P (V/Y). P (Y)$$

Applying the chain rule and using the dependency information represented in the network obtain this result. P (Y) is called the prior probability; and P (T/W), P (X/W), P (W/V, Y), P (U/V), and P (V/Y) are called the conditional probabilities. While, prior probabilities, probabilities based on initial information, can be obtained from statistical data using the relative frequencies, conditional probabilities can be elicited from experts or calculated using different types of mathematical models.

Within a belief network, the basic computation is to calculate the belief of each node (the node's conditional probability) based on the evidence that has been observed. This consists in instantiating the input variables, and propagating their effect through the network to update the probability of the hypothesis variables. An important purpose of BBNs is to facilitate calculation of arbitrary conditional probabilities. Various techniques have been developed for evaluating node beliefs and for performing probabilistic inference. The most popular methods are due to Pearl [8]. Similar techniques have been developed for constraint networks in the Dempster-Shafer formalism [8, 9].

We observed dependencies within the network dependency model in order to establish the weak and the strong influences among the variables in the model and to find the important variables for water quality. This procedure assists in forming some heuristics that will be cost-effective and useful not only for probabilistic inference but also for automatic construction of a belief network from data.

4 Applications of BBNs

Among more than forty wells in Salalah region, four wells only were selected to be analyzed. Those four wells have had, to the greatest extent, complete data measurements and provide sufficient information for the assessment of the groundwater quality for this selected basin. Another point worth to mention here is that all other wells in Taqah region are close to each other. We, therefore, ignored these wells because they add no additional information.

In our study, we only considered the dependencies between total dissolved solids (TDS), electrical conductivity (EC) and water pH. In the Sultanate of Oman, these are the main factors that expert in the area were dealing with and, therefore, maintained good data about them. Other factors, which are also considered less significant to groundwater quality in Oman, were not recoded and therefore neglected in this study.

TDS contains several dissolved solids but 90% of its concentration is made up of six constituents. These are: sodium Na, magnesium Mg, calcium Ca, chloride Cl, bicarbonate HCO3 and sulfate SO4. We used the following relationship between TDS and EC [1].

$$TDS = A * EC;$$ where A is a constant with value between 0.75 and 0.77.

Both TDS and EC can affect water acidity or water pH. Solute chemical constituents are variable in high concentration at lower pH (higher acidity). On the other hand acidity allows migration of hydrogen ions (H^+), which is an indication of conductivity. Therefore, our work concentrated on the following relations:

$$\begin{aligned} &TDS \rightarrow EC, \\ &EC \rightarrow pH, \\ &TDS \rightarrow pH. \end{aligned}$$

Knowing that the maximum allowable TDS in the drinking water is 600 mg/l, Table 2 shows the limits for a number of constituents of drinking water. The data sample is divided into two intervals (categories), considering TDS=550 is the central point. Thus, the first category is having TDS < 550 and the second category is having TDS >= 550. For EC, we also divide the data sample into two categories: data with EC < 670 and data with EC >= 670. Regarding pH, we also divided the data sample into two categories, data with pH < 7.5 and data with pH >= 7.5. The data table and the probability tables produced by this analysis for two wells are as follows:

Table 3 shows the monitoring measurements of the main components of TDS along with the measurements of EC and pH for Well 001/577. To analyze the relations mentioned above the following probabilities were calculated.

$$P\ (TDS < 550) = 0.556 \text{ and } P\ (TDS >= 550) = 0.444.$$

From the relationship between TDS and EC, the conditional probability presented in Table 4 was produced.

Table 5 shows the conditional probability table that shows the conditional probability of pH given TDS and EC. Similarly, we obtained the conditional probability tables for other wells.

Figures 2 and 3 show the data related to Well 001/580 that was treated by HUGIN [9]. Figure 3, for example, shows the HUGIN window which has two parts; the conditional probability table (CPT) P(EC/pH) and the network that represents the relationships between the variables TDS, EC and pH.

After providing the prior probabilities and the conditional probability tables, the results of the run session (probability update) for new-presented data for any selected node of HUGIN are shown in Figure 3.

We processed the data for other wells in the same way to build the BBN. We tested the BBN with different values of TDS, EC and pH taken from the collected data for the four wells that were selected for the development of this prototype.

Once the BBN model for each monitoring well has been built, parameterized and tested, these models can be interconnected in order to cover the whole basin. Figure 4 shows the BBNs for two monitoring wells.

Table 2. Drinking Water Standard

Element	Limit for Drinking Water
PH	7.0-8.5
Chloride mg/l	250
TDS	500-1000
Sulphate mg/l	200
Copper mg/l	1.3
Iron mg/l	0..5
Sodium	200-400

Table 3. Well 001/577

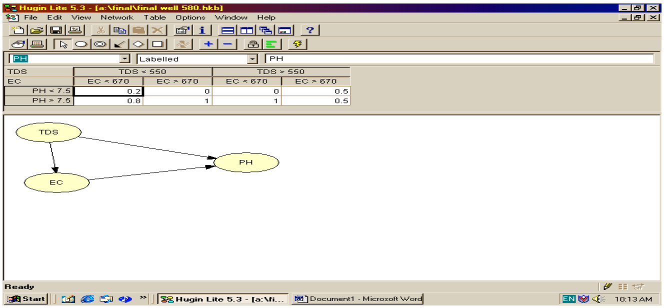
Mg	So4	Na	Ca	Cl	HCO3	TDS	EC	pH
17	14	31	96.11	51	186.9	440	671	7.85
7	11	20	60	35	124.4	286	386	7.4
16	25	28.5	52	60.91	156.8	377	491.25	8.2
12.62	19	30.3	101.65	51	314.15	587.5	741	7.3
11.75	13	32	31.8	62	114.7	295	430	7.7
13.57	15	24.69	98.31	55	336.6	603.5	726.5	7.3
14.4	20	27.7	104	59	310.2	595	668	7.9
15.5	23	32.2	135	88	360.5	727	827	7.4
12.4	17	28.1	115	67	315.6	617	716.25	7.8

Table 4. P(EC / TDS)

	TDS < 550	TDS >= 550
EC < 670	0.75	0.2
EC >= 670	0.25	0.8

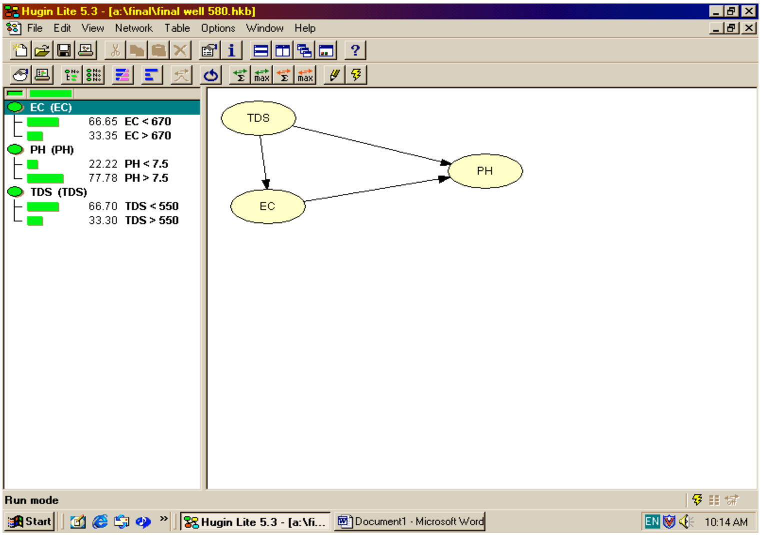
Table 5. P(pH/TSD, EC)

TDS >= 550		TDS < 550		Well 577
EC >= 670	EC < 670	EC >= 670	EC < 670	
0.667	0	0	0.333	pH < 7.5
0.333	1	1	0.667	pH >= 7.5


Hugin Lite 5.3 - [a:\final\final well 580.hkb]
File
Edit
View
Network
Table
Options
Window
Help
PH
Labelled
PH
TDS
EC
PH
Ready
Start
Hugin Lite 5.3 - [a:\fi...
Document1 - Microsoft Word
EN
10:13 AM

TDS	TDS < 550		TDS > 550	
EC	EC < 670	EC > 670	EC < 670	EC > 670
PH < 7.5	0.2	0	0	0.5
PH > 7.5	0.8	1	1	0.5

Fig. 2. A simple BBN representing the domain of groundwater quality, which is constructed using HUGIN.


Hugin Lite 5.3 - [a:\final\final well 580.hkb]
File Edit View Network Table Options Window Help
TDS
PH
EC
Start
Hugin Lite 5.3 - [a:\fi...
Document1 - Microsoft Word
EN
10:14 AM

EC (EC)
66.65 EC < 670
33.35 EC > 670
PH (PH)
22.22 PH < 7.5
77.78 PH > 7.5
TDS (TDS)
66.70 TDS < 550
33.30 TDS > 550

Run mode

Fig. 3. Probability update when EC>670

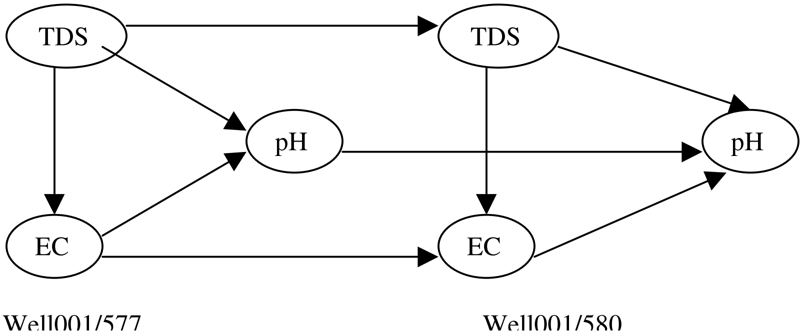


Fig. 4. BBN for two monitoring wells

5 Conclusion and Further Work

This work presents the assessment of groundwater quality. Bayesian Belief Networks (BBNs) have been investigated and shown to offer considerable potential for use in groundwater quality prediction. This technique is based on Bayesian methods for reasoning under conditions of uncertainty. BBNs present effectively the relationships between the constituents of the Water Quality. Therefore, the simple BBNs presented here is the first step towards having a comprehensive network that contains the other significant variables.

Data were collected from many monitoring wells in the Sultanate of Oman. We spent a significant time and efforts to have these data complete and useful for this study. We plan to continue this work to predict groundwater quality several months in advance, and in different parts of the area under study.

References

1. Lung Wu-Seng: Water Quality Modeling. CRC Press, Inc. (1993).
2. Anderson, M. and Woessner, W.: Applied Groundwater Modeling. Academic Press (1992).
3. Trigg, D. S. and at el: A prototype Bayesian belief network for diagnosis of acidification in Welsh rivers. Computer Techniques in Environmental Studies (2000) 163-172.
4. Chong H. G. and Walley W. J: Rule-based versus Probabilistic Approaches to the Diagnosis of Faults in Wastewater Processes. Artificial Intelligence in Engineering, Vol. 10 (1996) 265-273.
5. Varis, O.: Belief networks for modeling and assessment of environmental change. Environmetrics 6 (1995) 439-444.
6. Nicholson A. E. and Brady J. M.: Dynamic Belief Networks for Discrete Monitoring. IEEE Transactions on Systems, Man, and Cybernetics}, 24(11) (1994).
7. Russell, S., and Norvig P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Inc. (1995).
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann (1988).
9. HUGIN Expert Brochure, HUGIN Expert A/S, P. O. Box 8201 DK-9220, Aalborg, Denmark (2002).
10. Banerjee A. K. at el: TR no. 773, Department of Statistics, University of Wisconsin (1985).
11. Jensen, F. V.: Bayesian Networks and decision Diagrams, Springer (2001).

Facilitating E-learning with a MARC to IEEE LOM Metadata Crosswalk Application

Yang Cao, Fuhua Lin, Rory McGreal, Steve Schafer, Norm Friesen, Tony Tin, Terry Anderson, Doug Kariel, Brian Powell, and Margaret Anderson

Athabasca University
10030-107 Street Edmonton, AB. Canada T5J 3E4
{yangc, oscarl, rory}@athabascau.ca

Abstract. E-learning has the potential to provide the flexibility and wider access that is required for lifelong learning, but creating the digital resources needed for online course delivery requires a considerable investment and substantial effort. There are some pre-existing learning objects available for reuse in the design of educational events. However, supplying the metadata for each standard becomes repetitious, time-consuming, and tedious because of the diversity of learning objects, and the continuing growth in the number, size, and complexity of the content metadata standards. In order to minimize the amount of time needed to create and maintain the metadata and to maximize its usefulness to the widest possible community of users, there is a demand for developing crosswalks between different metadata standards. This paper introduces a crosswalk development that converts MARC (MAchine-Readable Cataloging) metadata to IEEE LOM (Learning Object Metadata) and investigates the issues involved in the crosswalk development.

1 Introduction

As the demand for access to education grows and an increasing numbers of adults return to universities/colleges for continuing education and training [1], so grows the need for new technologies to facilitate learning. E-learning provides great opportunities to increase flexibility in time and location of study, in terms of availability of information and resources, synchronous and asynchronous communication and various types of interaction [2] via the World Wide Web. Still, creating the digital resources needed for online course delivery requires considerable investment and effort. Reuse of learning objects is one optimal solution to address this problem.

According to the definition from the Learning Object Metadata Working Group of the IEEE Learning Technology Standards Committee (LTSC), a learning object is defined as "any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning"[3]. Learning objects vary in granularity ranging from as large as a chapter in a book, a case study, or an entire interactive course, to smaller items, such as a single pedagogical concept. In order for learning objects to

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 739-748, 2004.

be reusable, portable, and flexible, they are indexed with metadata so that they can be identified, located, accessed, retrieved, and assembled in the context of a particular task or learning activity.

In addition to the diversity of pre-existing learning objects, the number, size, and complexity of the content metadata standards continues to grow. Supplying the metadata for each standard becomes more repetitious, time consuming, and tedious. In order to minimize the amount of time needed to create and maintain the metadata and to maximize its usefulness to the widest community of users, there is a need for developing crosswalks among the metadata standards. Many projects [4, 5, 6] are underway to produce/collect learning objects and develop authoring tools for metadata generation. However, not much attention has been paid to the mapping from existing metadata standards to learning object metadata standards.

A crosswalk is a semantic and/or technical mapping (sometimes both) of one metadata framework to another metadata framework [7]. The purpose of this research is to, design and implement an automated tool for converting MAchine-Readable Cataloging (MARC) [8] to IEEE Learning Object Metadata (LOM) [9] and investigate the issues involved in the construction of this particular crosswalk.

2 MARC -> LOM/CanCore Converter

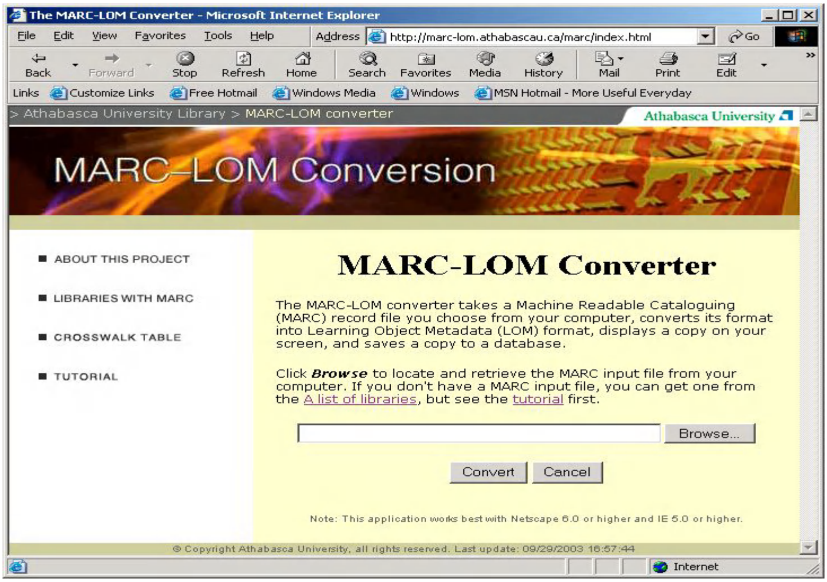
A MARC record contains bibliographic information, including a description of the item, subject heading, and the classification or call number, etc. The MARC record also contains a guide to its data by which computers can exchange, use and interpret bibliographic information and related information. Fig.1 shows a MARC record.

```
    00941nam  2200277 a
4504008004100000010001500041020001500056040002300071042000800094050002600010
2082001700128090002600145100003000171245006500201260004000266300002900306504
00510033565000250038665000350041165000260044665000180047285000120049091300
0600502990006700508995008800575-020114s2002    flum     b   a001 0 eng  -  -
a2002017488-  a0849310318-  aDLCbengcDLCdDLC-  apcc-00aQA76.9.D3bT4583 2002-00-
a005.75/8221-  aQA 76.9 .D3 T536 2002-1 aThuraisingham, Bhavani M.-10aXML databases
and the semantic Web /cBhavani Thuraisingham.-  aBoca Raton, FL :bCRC Press,c2002.-  -
a306 p. :bill. ;c24 cm.-  aIncludes bibliographical references and index.- 0aDatabase manage-
ment.- 0aXML (Document markup language)- 0aWeb site development.- 0aSemantic Web.-  -
aCaAEAU.
```

Fig. 1. An example of a MARC record

IEEE LOM is the most comprehensive schema developed for the description of the properties of learning objects. The LOM standard focuses on the minimal set of objects to be managed, located, and evaluated. CanCore [10] is a metadata implementation profile that is based on and fully compatible with the IEEE LOM standard and the IMS Learning Resource Meta-data specification. CanCore provides a set of guidelines to facilitate the implementation of the LOM. The converter that we have built can be called a MARC - LOM/CanCore converter because we have not treated CanCore and the LOM differently. They are the same, except that CanCore refines the LOM definitions and semantics.

The MARC-LOM/CanCore converter is a Web-based program written in Java that takes the metadata of library materials in MARC format and converts them into IEEE LOM format, displays a copy of the results on the user's screen, and saves the result to a LOM database. The interface of the converter is shown in Fig. 2.


The MARC-LOM Converter - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://marc-lom.athabascau.ca/marc/index.html
Go
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit
Links Customize Links Free Hotmail Windows Media Windows MSN Hotmail - More Useful Everyday
> Athabasca University Library > MARC-LOM converter
Athabasca University
MARC–LOM Conversion
ABOUT THIS PROJECT
LIBRARIES WITH MARC
CROSSWALK TABLE
TUTORIAL
Browse...
Convert
Cancel
Internet

MARC-LOM Converter

The MARC-LOM converter takes a Machine Readable Cataloguing (MARC) record file you choose from your computer, converts its format into Learning Object Metadata (LOM) format, displays a copy on your screen, and saves a copy to a database.

Click ***Browse*** to locate and retrieve the MARC input file from your computer. If you don't have a MARC input file, you can get one from the A list of libraries, but see the tutorial first.

Note: This application works best with Netscape 6.0 or higher and IE 5.0 or higher.

© Copyright Athabasca University, all rights reserved. Last update: 09/29/2003 16:57:44

Fig. 2. The interface of the MARC – LOM converter

2.1 System Design and Implementation

There are three components in the development of the MARC-LOM converter. The first component uploads MARC records (generated from a list of libraries' web sites) to the system. The second component is the MARC to MARC-XML conversion that converts the metadata in MARC format into MARC-XML architecture [11]. This component was developed based on part of the work of the Library of Congress [8]. The third component is the mapping from MARC-XML to LOM. Finally, the result in LOM format will be saved to the LOM database and a copy of the result will be displayed to the user. Fig. 3 illustrates the process of the conversion.

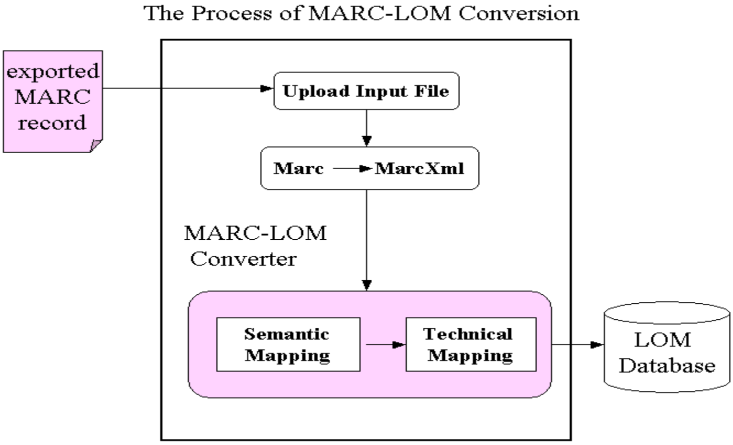


Fig. 3. The process of the conversion

The mapping from MARC-XML to LOM consists of semantic mapping and technical mapping. The semantic mapping is a mapping from each element in the source metadata standard to a semantically equivalent element in the target metadata standard. This process requires in-depth knowledge and specialized expertise in the associated metadata standards. The technical mapping involves a transformation that takes the metadata record content in MARC-XML format and translates it into a LOM record with compatible information. The technical mapping employs a set of computer programs and transformation languages. We use extensible Stylesheet Language Transform (XSLT) to programmatically change extensible Markup Language (XML) metadata records from MARC-XML format to LOM XML format. The following example shows MARC-XML elements "language" and "title" and their corresponding LOM XML elements.

- MARC-XML elements

```
<controlfield tag="008">020114s2002    flum     b   a001 0 eng  </controlfield>
 <datafield tag="245" ind1="1" ind2="0">
     <subfield code="a">XML databases and the semantic Web /</subfield>
 </datafield>
```

- LOM XML elements

```
 <language>en</language>
 <title>
 <string language="en">XML databases and the semantic Web  </string>
 </title>
```

It is intelligible if MARC tag is 245a then it means "title" in LOM. But it might confuse readers that how to obtain the information on "language" of the title from the element "title" in MARC. In fact, this information is obtained from an element "language" other than from "title" in MARC. The above example illustrates that the differences in terminology and structures existed in two metadata standards and the mapping needs “semantic” mapping not just simple element or structure mapping.

The mapping is an iterative process involving ongoing refinement, revision, and, even rethinking of element matches. The challenges in the semantic & technical mapping were the lack of a common terminology, different structures of metadata standards, lack of expertise in element mapping, and the loss of information [12]. Furthermore, maintaining the crosswalk as the metadata standards change becomes even more problematic due to the difficulties in keeping a historical record, and anticipating and responding to new changes in the associated standards. Some of the problems encountered in the crosswalk development include:

Lack of common terminology **--** A lack of common terminology currently exists among the different metadata standards. Normally, this means that the same thing is described by different terms in various metadata standards. Another case of this is that the same names do not mean the same thing in heterogeneous metadata standards. For example, size of the library material might refer to the number of pages of a book, whereas in LOM the “size” field requires information about the size of learning object in bytes. Therefore, it should not be a revelation to find that a particular element in a metadata standard is not equivalent to an element in another metadata standard even though they have the same name.

Different structures of metadata standards **--** Each metadata standard is organized differently since normally they developed independently. Different structures of metadata standards make it difficult to find corresponding information in analogous sections of another metadata standard.

Expertise in element mapping –- There are three situations in the element to element technical mapping: one to one; one to many; and many to one. In most cases, a one-to-many map is trivial, because an occurrence of the source element can map to a single occurrence in the target element. For example, a single element “language” in MARC is simply mapped with the element “language of title” and “language of keyword” in LOM. However, sometimes mapping one source element to two or more unique target elements requires specialized knowledge on how to interpret the source element and parse it to corresponding target elements. An example of this case is when the name of a person or an organization in MARC is mapped to the same element in LOM. In LOM, the name of a person or an organization is expressed by VCARD in which first name, last name, middle name, and title of a person are separated by a different delimiter. This requires that developers parse the information in a single element “name” to units including first name, last name, etc and reorganize these units into LOM.

The many-to-one mapping must specify what to do with the extra elements. For example, there is more than one value for the title in MARC, such as title, subtitle, alternative title, etc. On comparison, there is only one value for title in LOM. If the resolution is to map all values of the title in MARC to a single value in the LOM, explicit rules are required to specify how the values will be appended together.

Loss of information **--** Each metadata standard was built for a different purpose, so it is not surprising to find that the number of elements in each metadata standard is different. There are always some elements for which we could not find a compatible match in the other metadata standard. For example, the element "Physical Description" in MARC (tag 300) couldn't match with any corresponding element in LOM. Loss of information is unavoidable.

2.2 Research Status

The MARC-LOM Converter has been fully implemented at Athabasca University and it is house on the AU web server (http://marc-lom.athabascau.ca/marc/). The MARC-LOM crosswalk table was created and evaluated by the AU team with expertise in library science, learning object metadata standards (LOM & CanCore) and computer technologies. Appendix A shows a simple version of the cross walk table. Appendix B shows a LOM result that was generated from the MARC-LOM converter. The full crosswalk table and result can be found at the MARC-LOM web site.

We conducted some preliminary experiments with users from the library and university centers. The library records they used are from the Athabasca University Catalogue. In addition to the AU Catalogue, we have tested the MARC-LOM/CanCore Converter with another15 libraries around the world (see appendix C) and the converter worked as expected. The MARC-LOM Converter and other information about the project is available at URL http://marc-lom.athabascau.ca/marc/ .

3 Conclusions and Future Work

The MARC-LOM/CanCore Converter allows MARC based library content metadata and harvested metadata from the library system to interact with XML based digital repositories. The converter enables the simplification of the creation/maintenance of metadata and their utilization in e-Learning. It will render re-purposable knowledge objects readily available alongside specially purposed learning objects and bring us one step closer to enabling the use of intelligent agents on the semantic web. Our future work will focus on the aspects of interoperability of functions including finding resources and retrieving them, putting objects into repositories, and supporting interactions between repositories. Heterogeneous software agents, such as converting agents, searching agents, and monitoring agents will be employed to maintain robust networks of learning object repositories.

Acknowledgement. This project is funded by Industry Canada and was part of an initiative called IMEC (Interoperable Metadata for e-Learning in Canada) as well as the pan-Canadian eduSource project funded by CANARIE, Canada's broadband research network. The software application was based partially on the work of the Library of Congress.

References

1. CIHE: The Council for Industry and Higher Education, Response to the joint consultation document from HEFCE and the Learning and Skills Council (2002), http://www.cihe-uk.com/partnershipsfor.htm
2. Cao, Y., Jim, G.: Agent Programmability in a Multi-Agent Learning Environment. Proceeing
3. of the 11th International Conference on Artificial Intelligence in Education. Australia(2003).
4. Learning Technology Standards Committee (LTSC): http://ltsc.ieee.org/
5. CETIS Group: http://www.cetis.ac.uk/
6. MERLOT: http://www.merlot.org/Home.po
7. Wisconsin Online Resource Center - Wisc-Online Learning Object Project: http://www.wisc-online.com/
8. Digital Library for Earth System Education (DLJESE): http://www.dlese.org/Metadata/crosswarlks
9. MAchine-Readable Cataloging (MARC): http://www.loc.gov/marc/
10. IEEE Learning Object Metadata (LOM): http://ltsc.ieee.org/doc/wg12/LOM-WD3.htm
11. CanCore Initiative 2003. http://www.cancore.org
12. MARC-XML architecture http://www.loc.gov/standards/marcxml/marcxml-architecture.html
13. St. Pierre, M., LaPlant, W. P. Jr. : Issues in Crosswalking Content Metadata Standards (1998) http://www.niso.org/press/whitepapers/crsswalk.htm

Appendix A: MARC 21 to LOM/CanCore Crosswalk Table

LOM/CANCORE ELEMENT	MARC TAG AND SUBFIELD	MARC DESCRIPTION	CONDITIONS
1.1 Identifier	020 a	ISBN	If MARC tag is 020, then the catalog = ISBN
	022 a	ISSN	If MARC tag is 022, then the catalog = ISSN
	856 u	Electronic Location and Access	
1.2 Title	245 a	Title	
	245 b	Remainder of title	
	130 a	Main entry Uniform title	
	240 a	Uniform title	
	630 a	Subject uniform title	
	246 a	Varying forms of title	
1.3 Language	008/35-37	Language	
	041 a	Language	
1.4 Description	520 a	Summary	
1.5 Keyword	600 a	Subject – Personal	
	610 a	Subject – Corporate	

	650 a	Subject – Topical	
	611 a	Subject – Conference	
2.1 Version	250 a	Edition	
2.3 Contribute 2.3.1 Role	100 a	Personal author	Field present if item was created by the entry.
	110 a	Corporate author	Same as 100
	008 byte 22	Target audience	Only valid if item is book, computer file, music, or visual material
: : :	: : :	: : :	: : :
9.2.1 Source	600, 610, 611,650, 651	Subject	
	050 or 055 or 082 All subfield a	LC Classification number Dewey Decimal Classification number	If 050 or 055 field is present, the classification is LC in type. 082 indicates DDC
	090	Local call number	If 2-3 letters and then numbers, LC Class is Source. (AB 2321) If entry is solely numbers (111.11….), then Source is DDC.
	099	Local call number	
9.4 Keyword	600, 610, 611, 650, 651	Depends on the actual MARC field	

Appendix B: A Sample LOM Result

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
-<lom xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
     xsl:schemaLocation="http://ltsc.ieee.org/xsd/LOMvlp0
     http://adlib.athabascau.ca/catalog/xml/LOMv1p0/lom.xsd"
     >
  - <general>
     - <identifier>
          <catalog>ISBN</catalog>
          <entry>0849310318</entry>
       </identifier>
     - <title>
          <string language="en">XML databases and the
             semantic Web</string>
       </title>
       <language>en</language>
```

```
    - <keyword>
        <string language="en">Database management.</string>
      </keyword>
    - <keyword>
        <string language="en">XML (Document markup
            language)</string>
      </keyword>
    - <keyword>
        <string language="en">Web site development.</string>
      </keyword>
    - <keyword>
        <string language="en">Semantic Web.</string>
      </keyword>
    </general>
  - <lifeCycle>
    - <contribute>
      - <role>
          <source>LOMv1.0</source>
          <value>Author</value>
        </role>
      - <entity>
          <vcard>BEGIN:VCARD VERSION:3.0 FN:Bhavani
              M.Thuraisingham N:Thuraisingham; Bhavani; M.
              END:VCARD</vcard>
        </entity>
      </contribute>
    - <contribute>
      - <role>
          <source>LOMv1.0</source>
          <value>Publisher</value>
        </role>
      - <entity>
          <vcard>BEGIN:VCARD VERSION:3.0 ORG:CRC Press
              END:VCARD</vcard>
        </entity>
      - <date>
          <dateTime>2002</dateTime>
        </date>
      </contribute>
    </lifeCycle>
  - <technical>
      <format>non-digital</format>
      <location>QA 76.9 .D3 T536 2002</location>
    </technical>
    <right />
  </lom>
```

Appendix C: A List of Online Public Access Catalogues Compatible with the MARC-LOM Converter

Australia
Murdoch University, University of Sydney, University of Western Australia, University of Wollongong

Canada
Athabasca University Library, Simon Fraser University, University of Winnipeg, Canada Institute for Scientific and Technical Information

Taiwan
Taiwan National University, Academia Sinica

United States
Brown University, San Diego State University, University of Missouri
University of Massachusetts at Amherst, University of Nebraska, Omaha

An Agent-Based Framework for Adaptive M-learning

Larbi Esmahi [1] and Elarbi Badidi [2]

[1] Center for Computing and Information System, Athabasca University,
1 University Drive, Athabasca, Alberta, T9S 3A3, Canada
larbie@athabascau.ca
Tel: +1 780 6756657, Fax: +1 780 6756186
[2] Dpartement de Biochimie, Universit de Montral
C.P. Succursale Centre-ville, 2900 Blvd Eduard-Montpetit
Montréal, Québec, H3C 3J7, Canada
me.badidi@umontreal.ca
Tel: +1 514 343-6111 (5188), Fax: +1 514 343-2210

Abstract. This paper describes a multiagent system for delivering adaptive m-learning services. It provides a discussion of many questions related to adaptivity and mobility in e-learning: course content adaptation, wireless access to learning services using PDA, and openness to external learning resources (learning object repositories). The paper provides a short overview of some related works and provides a detailed description of the architecture and components of the proposed multiagent framework.

1 Introduction

Electronic learning continues to grow phenomenally but most e-learning development involves wired infrastructures. It is believed that emerging wireless and mobile networks will provide new applications in mobile learning [1]. Learners can access mobile learning services from anywhere, anytime, and even under varying degrees of network failure. Web-based learning systems development has taken two different but complementary directions. On one hand research-based Intelligent Tutoring Systems (ITS) [2] and Adaptive Hypermedia Systems (AHS) [3] that focus on the problem of adapting the instructional process (course content adaptation, course navigation adaptation, problem-solving support, etc.). These systems are underused and always don't go beyond their research environment. On the other hand industrial-based Learning Management Systems [4] whose primary focus is the management of the learning process (Registration and tracking of students, Content creation and delivery capability, Skill Assessment and development planning and Organizational resource management). These systems are largely adopted in the education and training market.

This paper presents an M-learning framework whose main objective is to bridge the gap between the modern approach to Web-based education (based on mobile devices, wireless networks and learning object repositories), and powerful but underused ITS and AHS technologies. This framework attempts to address both the

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 749-758, 2004.

component-based development of adaptive systems, the user mobility and the openness toward external course content providers and learning object repositories [5].

In section 2 we present some pertinent literature and related works. Section 3 presents the architecture of the proposed framework. Finally we conclude with some comments about future developments related to this work.

2 Literature Overview and Related Works

In the literature, M-learning has been defined from different views. Some definitions take technology as the starting point [6, 7], other definitions [8] relate it more to distance education by focusing on the principle of anytime, anywhere and any device. Leung [9] identifies four characteristics for m-learning: dynamic by providing up-to-date material and resources, operating in real time by removing all constraints on time and place, adaptive by personalizing the learning activities according to the learner background and collaborative by supporting peer-to-pear learning. M-learning is still in its birth stage and most of the research projects are focusing on the connectivity problem of using wireless networks or the problem of accessing course content using mobile terminals (PDAs such as Compaq iPaq or WAP phones) [10, 11]. The multi-device issue has been also addressed in some projects. Ketamo et al. [12] have implemented an m-learning environment (xTask) that adapt to different user devices (PC, PDA and WAP devices). xTask implements also a library for managing learning material in various formats and discussion forum tools. In the Knowmobile project Gallis et al. [13], studied how medical students use various information and communication devices in the learning context and argue that " there is no 'one size fit all' device that will suite all use situations and all users. The use situation for the medical students, points towards the multi-device paradigm". The multi-device paradigm fits well with the e-learning context where students use different devices depending on the situation, environment and context.

Few of the M-learning projects have addressed the problems of adaptation of learning tasks and personalization of course content based on student's model, learning styles and strategy [14, 15]. These problems have been largely studied within the traditional web-based systems (ITS and AHS). Some of the well-known projects are: ELM-ART [16, 17] an on-site intelligent learning environment that supports example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. InterBook [18,19] a tool for authoring and delivering adaptive electronic textbooks. It supports adaptive sequencing of pages, adaptive navigation by using links annotation and adaptive presentation. DCG [20, 21, 22] an authoring tool for adaptive courses. It supports adaptive sequencing and offers different levels of re-planning the course. AHA [23, 24, 25, 26] a generic system for adaptive hypermedia whose aim is to bring adaptivity to all kinds of web-based applications. AHA supports adaptive navigation (annotation + hiding) and adaptive presentation. ILESA [27, 28] an intelligent learning environment for the Simplex Algorithm. It implements adaptive sequencing (Lesson, problem) and provide problem-solving support.

Personalization of the teaching material has been studied and evaluated in the era of psychology of learning and teaching methods [29, 30, 31]. The empirical evaluation of these methods showed that personalized course material increases the learning speed and help learners for better understanding [14].

At least three major technical issues should be considered when building m-learning services: how to deal with unreliable and low bandwidth of wireless connections using handheld devices, how to insure a seamless multidevice and cross-platform solution, and how to manage learning services adaptivity to the learner's preferences and style.

In this paper we are proposing a solution based on mobile agents and a two-fold student modeling mechanism. Mobile agents will allow us to address the problem of intermittent connectivity and multidevice / cross-platform adaptation and the two-fold student modeling will allow us to handle in a very flexible manner the learning services personalization.

3 Multiagent Framework for Adaptive M-learning

3.1 Context

The design basis for this framework is a university e-learning environment (i.e. Athabasca University) where learners are taking courses in asynchronous mode (grouped or individual). In this context most of learners are working and have more personal needs to enhance their careers. They often have a long term learning plan, and prefer flexible and individualized environment. A significant percentage of these learners are always mobile and need to have access to their courses from everywhere, anytime and using different devices. So, the context of this project takes into consideration the following requirements:

- Asynchronous e-learning.
- Mobile users.
- Multi-devices environment.
- Adapted courses.
- Personalized interfaces that offers the same look and feel.

3.2 System Architecture

In this context we are proposing a multiagent architecture for implementing an e-learning system that offers courses personalization and support mobile users connecting from different devices. The detailed architecture of the system is articulated around five main components:

1. Users profiles repository: for each user the system maintains a profile that has two components, the learner's model and the user preferences regarding the learning style, interfaces and content display.

2. Devices profiles repository: contains for each device the features and capabilities useful for the e-learning service provision (screen size, bandwidth limit, colors, resolution, etc.). Some features that can be automatically detected by the system (Operating System, Browser, Plug-ins) are not stored in the repository but integrated to the profile when initializing the terminal agent.
3. Learning objects repository: contains the courses teaching material defined as learning objects [5].
4. Courses database: for each course the system maintains two knowledge structure: the course study guide and the course study plan.
5. Multiagent system composed of stationary and mobile agents.

The following figure presents the main components of the system:

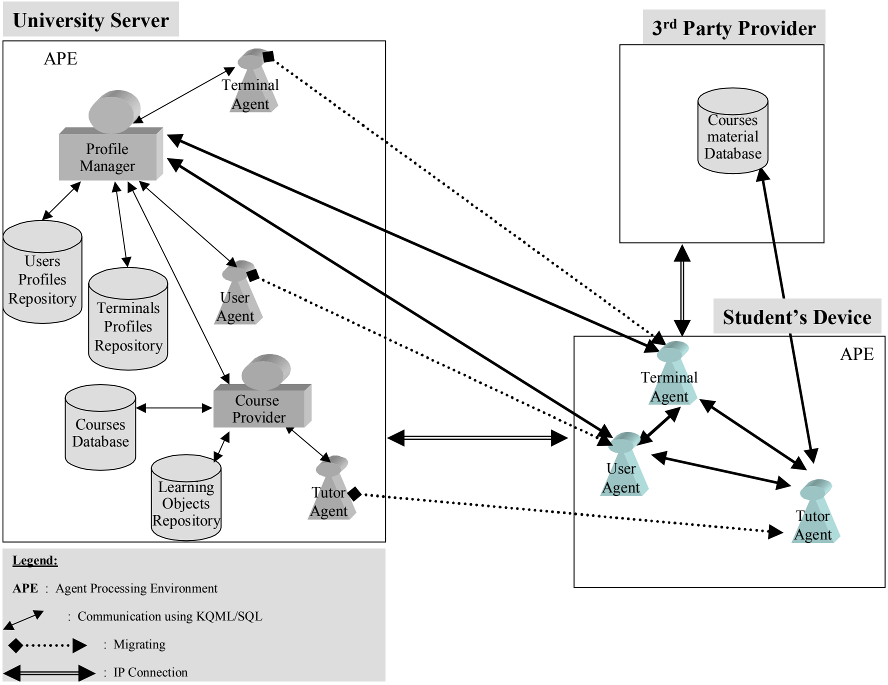


Fig. 1. System Architecture

Profile Manager Agent: the profile manager is implemented as a stationary agent. It manages the knowledge related to the learners and all defined devices (terminals). The main tasks of the profile manager are:

- Performing user authentication
- Acting as a central register, where each new learner must be registered.
- Managing and assuring the consistence of the databases containing learners and devices profiles.
- Receiving service requests from terminals and giving access to user profiles data.

- Initiating and sending the user agent and terminal agent to the remote device.
- Checking the version of the user agent and the terminal agent that resides on remote terminals and automatically download any necessary updates.

Course provider Agent: a stationary agent that manages the knowledge about courses and teaching strategies. The main tasks of the course provider are:

- Providing an interface for defining learning objects and courses knowledge (study guide and study plan).
- Receiving service requests from terminals and giving access to courses data.
- Generating the course study guide and study plan based on the user profile and the teaching strategy.
- Packaging the course teaching material according to the user profile and device profile.
- Initiating and sending the tutor agent and terminal agent to the remote device.

User Agent: a mobile agent that carry and manage a local copy of the user profile (user preferences and learner's model) to the remote terminal. The main tasks of the user agent are:

- Providing the tutor agent and terminal agent with the user information (profile, identification).
- Managing and synchronizing the user profile duplication with the central server.
- Providing the local personalization of the course material. In collaboration with the terminal agent and the tutor agents, the user agent insures the display of the course material according to user preferences and terminal capabilities.

Tutor Agent: a mobile agent that manage the course delivery to the roaming user. The main tasks of the tutor agent are:

- Carry and manage the course material and study plan.
- Provides a personalized learning service to the learner based on his model and learning style.
- Insure the adaptation and packaging of external course content. Since the system is open to third party providers, external course material will need to be converted to the required format and adapted to the user and device profile. The tutor agent insures the necessary adaptation and conversion in collaboration with the user agent and the terminal agent.
- Synchronizes the course content with the server.

Terminal Agent: a mobile agent that maintains the terminal profile of the corresponding device and insure the display of services according to the user preferences and terminal capabilities.

3.3 Knowledge Structures for Adaptive Courses Delivery

Course's study guide: the course content is organized around a set of concepts. Each concept has a teaching material associated with it that comes either from an external source or learning object. The course study guide defines the relationships between these concepts. The relationships consist of prerequisite, similarity and substitute relationships.

Course's study plan: The course structure is organized in terms of units and sections, and for each section a set of concepts is learned using different tasks: readings, labs and tests. The study plan defines the sequencing of the course content and the time constraints and deadlines for different tasks of the learning process.

Learner's model: a fuzzy overlay model based on the course concepts. It represents static beliefs about the learner and in some case is able to simulate the learner's reasoning. With each concept in the model is associated a fuzzy value that represents the assessment of the learner's knowledge regarding this concept.

Two different versions of the learner's model are used by the system: a global model and a local model. The global learner's model is stored within the user profile repository and represents concepts reported to the system about the learner or learned from different courses. This model represents in addition to the concepts and associated fuzzy values, the relationships between concepts (prerequisite, similarity and substitute). The local learner's model is managed by the user agent within the user's terminal and is related to a specific course. This model represents only the course concepts and associated values.

The local learner's model is refined based on the learner's interaction with the system when reading the course material and doing the assessment exercises. The local model is also used to update the global learner's model. The local model is initialized from the global model.

Teaching Strategy: A set of rules that implement the adaptation controls for the courses. It consists of rules for sequencing the course material components, rules for adding or dropping course material components and rules for selecting between similar or equivalent course material components

3.4 Adaptation of Third Party Provider Content

Unlike traditional systems, in this project we seek to be open to third party providers. In fact we aim in the future to implement an infrastructure (e-market place) for collaborative e-learning services provisioning. So, we need to implement a process that provides user-side device independence on web content. The main idea behind this process is to construct a basic generic page from the source, and then mark up that document with appropriate tags as determined by the user's profile and device profile.

A Web course content always involves different resources (files, database, learning objects… etc). So, the adaptation process consists of creating a Java Servlet or JSP document that connects to data sources and objects, and produces an XML document. The main idea here is to use a two-stage process that generates in the first step an XML document (model), and then translate the generated model to a rendering format (HTML, WML, etc.) that will be presented to the user.
The following figure describe the two stage services:

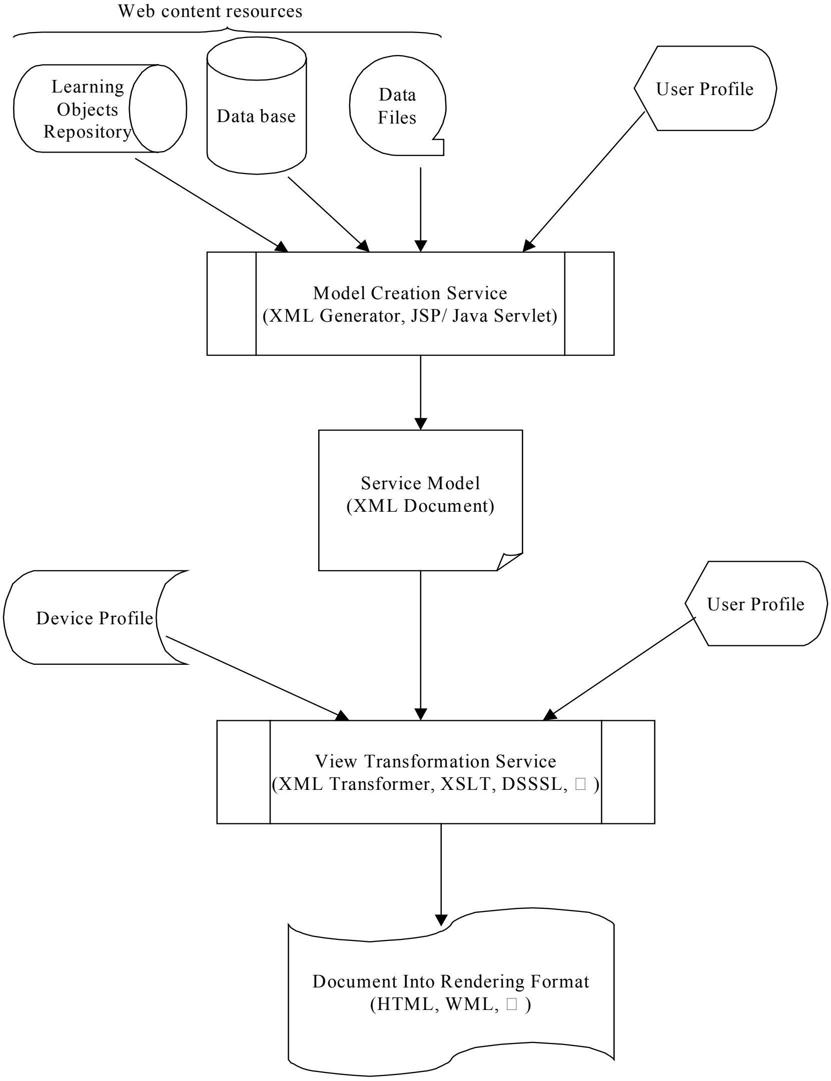

Web content resources
Learning Objects Repository
Data base
Data Files
User Profile
Model Creation Service (XML Generator, JSP/ Java Servlet)
Service Model (XML Document)
Device Profile
User Profile
View Transformation Service (XML Transformer, XSLT, DSSSL, □)
Document Into Rendering Format (HTML, WML, □)

Fig. 2. Adaptation of external course content

The first step in the content module is to create an XML document from the content resources. If we use the logic of web services this will correspond to the model creation service (figure 2). This model creation service implements an XML generator using JSP or Java Servlet that generate an XML document of the Web content. The user profile is used here to personalize the content according to the user profile. The second step in the content module is to create a rendering format for the XML document. This corresponds to the view transformation service (figure 2). The view transformation service implements an XML transformer using XSLT or DSSSL that generate a rendering format for the XML document. Since the rendering format depends on the device features and user preferences, the user profile and device profile will be used in this process.

The two stage services will provide us with more flexibility and device independence:

- The separation of the service model from the service view will provide us with device independence and make easy the maintenance of the content generation process.
- With browsers including a W3C-Compliant XSLT engine, more processing will occur on the client side and reduce the work done by the server.
- The services may be distributed over several machines if needed to balance the overall load.

4 Conclusion

The proposed multiagent architecture offers the advantage of being more flexible and scalable. The two-fold learner's model offers more flexibility to do the course adaptation either in the server or the client depending on the needs. The mobile agents architecture offers also a dynamic adaptation either for the course content or the interface since the tutor agent is in permanent communication with the user agent and the terminal agent. Furthermore the system is open to third party providers for the course content.

Future developments on this project will be focused on both solving some technical issues related to the implementation (complex knowledge structure within mobile agents) and extending the system toward an infrastructure (e-market place) for collaborative e-learning services provisioning.

References

1. Gayeski,,D.: Learning Unplugged: Using Mobile Technologies for Organizational Training and Performance Improvement, AMACOM, New York (2002).
2. Brusilovsky, P.: Intelligent tutoring systems for World-Wide Web. In R. Holzapfel (Eds.), Poster proceedings of Third International WWW Conference. Darmstadt, (1995), 42-45.
3. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction, Vol. 6 (2-3), (1996) 87-129.

4. Esmahi L., Holt P., Wang H. & Abaza M.: An essay in e-learning tools categorization. In the Proceedings of the 5th IASTED International conference on Computers and Advanced Technology in Education (CATE'02). Mexico (2002) 506-509.
5. Wiley, D. A.: Connecting learning objects to instructional design theory. A definition, a metaphore, and a taxonomy. In D. A. Wiley, (eds.), The Instructional Use of Learning Objects (2000), Online Version. Retrieved August 20, 2003, from the World Wide Web: http://reusability.org/read/chapters/wiley.doc
6. Quinn C.: mLearning: Mobile, Wireless, In-Your-Pocket Learning. LiNE Zine (2000-2001). Retrieved August 20, 2003, from http://www.linezine.com/2.1/features/cqmmwiyp.htm
7. Farooq U., Schafer W., Rosson M.B. & Carroll J.M.: M-Education: Bridging the Gap of Mobile and Desktop Computing. IEEE International Workshop on Mobile and Wireless Technologies in Education (WMTE'02). August (2002).
8. Nyiri J.C.: Toward a philosophy of M-Learning. IEEE International Workshop on Mobile and Wireless Technologies in Education (WMTE'02). August (2002).
9. Leung C.H. & Chan Y.Y.: Mobile Learning: A New Paradigm in Electronic Learning. Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT '03), (2003).
10. Houser C., Thornton P. & Kluge D.: Mobile Learning: Cell Phones and PDAs for Education. In IEEE Proceedings of the International Conference on Computers in Education (ICCE'02), (2002).
11. Baek Y.K., Cho H.J. & Kim B.K.: Uses of learning objects in a wireless Internet based learning system. In IEEE Proceedings of the International Conference on Computers in Education (ICCE'02), (2002).
12. Ketamo H.: xTask — Adaptable Working Environment. Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02), (2002).
13. Gallis, H., Kasbo, J. P. & Herstad, J.: The multidevice paradigm in know-mobile - does one size fit all?. In S. Bjørnestad, R. E. Moe, A. I. Mørch & A. L. Opdahl (Eds.), Proceedings of the 24th Information System Research Seminar in Scandinavia, (2001) 491–504.
14. Brusilovsky, P.: Adaptive navigation support in educational hypermedia: The role of learner knowledge level and the case for meta-adaptation. British Journal of Educational Technology, Vol. 34 (4), (2003) 487-497.
15. Esmahi L. & Lin F.: A multiagents framework for an adaptive e-learning system. Book chapter, to appear in "Designing Distributed Learning Environments with Intelligent Software Agents", Lin Fuhua, Editor, Idea Group Inc. (IGI), (2004).
16. Weber, G. & Specht, M.: User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems. In A. Jameson, C. Paris & C. Tasso (Eds.), Proceedings of the Sixth International Conference on User Modeling, UM97. Vienna, New York: Springer Wien (1997) 289-300.
17. Weber, G. & Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, (2001) 351-384.
18. Brusilovsky, P., Eklund, J. & Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, Vol. 30 (1-7), (1998) 291-300.
19. Brusilovsky, P. & Eklund, J.: A Study of User Model Based Link Annotation in Educational Hypermedia. Journal of Universal Computer Science. Vol. 4 (4), (1998) 429-448.

20. Vassileva J.: Dynamic Courseware Generation on the WWW. In A. Jameson, C. Paris & C. Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97 Vienna, New York: Springer Wien (1997) 433-435.
21. Vassileva J. & Deters R.: Dynamic Courseware Generation on the WWW, British Journal of Educational Technologies, Vol. 29(1), (1998) 5-14.
22. Brusilovsky, P. & Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning Vol. 13 (1-2), (2003) 75-94.
23. De Bra, P., Aerts, A., Houben, G. J. & Wu, H.: Making GeneralPurpose Adaptive Hypermedia Work. In the proceedings of WebNet—World Conference on the WWW and Internet, Association for the Advancement of Computing in Education, (2000) 117-123.
24. De Bra, P. & Ruiter, J. P.: AHA! Adaptive Hypermedia for All. In the proceedings of WebNet—World Conference on the WWW and Internet, Association for the Advancement of Computing in Education, (2001) 262-268.
25. De Bra, P., Aerts, A., Smits, D. & Stash, N.: AHA! Version 2.0, More Adaptation Flexibility for Authors. In the Proceedings of the e-Learn—World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Association for the Advancement of Computing in Education, (2002) 240-246.
26. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D. & Stash, N.: AHA! The Adaptive Hypermedia Architecture. In the Proceedings of the ACM Hypertext Conference, ottingham, UK (2003). Retrieved August 20, 2003, from http://wwwis.win.tue.nl/~debra/ht03/pp401-debra.pdf
27. López, J. M, Millán, E., Pérez J. L. & Triguero F.: Design and implementation of a web-based tutoring tool for Linear Programming problems. In Proceedings of workshop Intelligent Tutoring Systems on the Web at ITS'98, 4th International Conference on Intelligent Tutoring Systems (1998). Retrieved August 20, 2003, from http://www.lcc.uma.es/~eva/investigacion/papers/its98wsh.ps
28. López, J. M, Millán, E., Pérez J. L. & Triguero F.: ILESA: A web-based Intelligent Learning Environment for the Simplex Algorithm. In C. Alvegård (eds.), Proceedings of the 3rd International Conference on Computer Aided Learning and Instruction in Science and Engineering CALISCE'98 (1998). Retrieved August 20, 2003, from http://www.lcc.uma.es/~eva/investigacion/papers/ilesa.ps
29. Tennyson, R. D. & Christenson, D. L.: MAIS: an intelligent learning system. In D. H. Jonassen (Eds.), Instructional Designs for microcomputer courseware. Hillsdale: N. J.: Erlbaum (1988).
30. Litchfield, B. C., Driscoll, M. P., & Dempsey. J. V.: Presentation sequence and example difficulty: Their effect on concept and rule learning in computer-based instruction. Journal of computer-based instruction, Vol. 17, (1990) 35-40.
31. Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (Eds.), Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, Vol. 4, (1999) 19-25.

Determination of Learning Scenarios in Intelligent Web-Based Learning Environment

Elżbieta Kukla, Ngoc Thanh Nguyen, Janusz Sobecki, Czesław Daniłowicz, and Mateusz Lenar

Wrocław University of Technology, Department of Information Systems
Wybrzeże St. Wyspiańskiego 27, 50-370 Wrocław, Poland
{e.kukla,thanh,sobecki,danilowicz,lenar}@pwr.wroc.pl

Abstract. The paper presents a method of learning scenarios determination in intelligent web-based learning environment. The scenario is defined as a sequence of concepts from domain knowledge represented by a series of hypermedia pages. Presentation techniques used for pages construction correspond with teaching methods that are applied to the class of students characterized by certain definite learning style. Every student taking up a course is represented by his profile. For a given student and course the opening learning scenario is determined by consensus-based procedure on the ground of the similar learners scenarios. As the learning process proceeds the scenario is changed dynamically to better suit actual learner characteristics.

1 Introduction

Today the need for life-long education is obvious for everyone. It is addressed not only for pupils or students but also for staff members of various businesses, legal, governmental, educational and many other types of organizations. The increasing unemployment in many countries also speeds up the need for gaining new competences.

In this context e-Learning acquires larger and larger importance. It provides the possibility of learning at any time and anywhere. In order to enable an easy and open access to e-Learning courses they are usually implemented using the Web-based technologies, which are inexpensive but very efficient tools for delivering education and training for many different audiences [4].

In order to attract many participants e-Learning should offer its users adaptive, dynamically changing, intelligent environment that would contribute to improve the learning efficiency. Modern pedagogical theories as constructivism, cognitive apprenticeship, minimalism, conversational instruction and others provide the indications for systems practical design and solutions of these problems [3]. Most of them focuses on learning rather than teaching and attach great importance to learners as active constructors of their knowledge.

The approach proposed in this paper is based on user centred environment and intelligent guidance. It would provide adaptive guidance for the student in hypermedia structured learning space. This solution permits to determine optimal

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 759-768, 2004.

learning scenarios for different classes of students according to their personal features (e.g. age, completed education etc.), cognitive characteristics represented by individual learning style and sequences of hypermedia pages (presentations) suitable to teaching methods available for every piece of knowledge to be learnt.

2 Outline of Knowledge Structure and Learning Process

A learning scenario represents the proceeding of individual student's learning process. It indicates an order of knowledge items to be learnt and teaching methods used for their presentations [10]. Individualization of learning process relies on matching a scenario for a student according to learner's characteristics and predetermined knowledge structure [8].

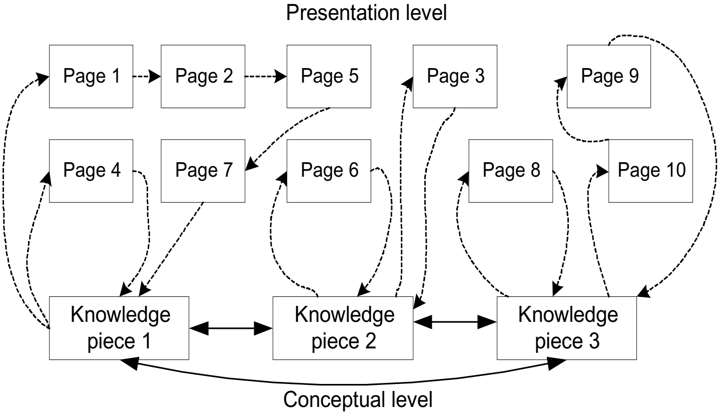


Fig. 1. Outline of Knowledge Structure

Let K be the finite set of knowledge units resultant from the whole knowledge domain partition on elementary, indivisible and meaningful pieces that are connected each other by different relations (respectively nodes and arcs at conceptual level on Fig. 1). We assume that there is defined a partial order α in K, which represents the obligatory order in learning these knowledge elements. By R_k we denote the set of all presentations of a knowledge unit $k \in K$. Presentations are the sequences of hypermedia pages related to every node from conceptual level (presentation level on Fig. 1). Each sequence corresponds to a teaching method suitable to the knowledge unit presentation. By T_k we denote the set of tests serving to verify learner competence referring to k. A presentation from R_k and a test from E_k are called *corresponding* to k. Let $\boldsymbol{R} = \bigcup_{k \in K} R_k$ and $\boldsymbol{T} = \bigcup_{k \in K} T_k$. A course L can be defined as a subset of knowledge units, that is $L \subseteq K$. Let $\alpha_L = \{(k,k'): k,k' \in L \text{ and } (k,k') \in \alpha\}$.

Definition 1. By a learning scenario s for course L we call a sequence belonging to Cartesian product $(R \cup T)^{2n}$ for n being the cardinality of set L, which fulfils the following conditions:

a) Each knowledge unit $k \in L$ has exactly 1 corresponding presentation and 1 corresponding test in s and s does not contain any other presentations and tests corresponding to knowledge unit k.

b) The order of any 2 presentations in s should correspond to the order (if exists) of their knowledge units in relation α_L.

c) The order of any 2 tests in s should correspond to the order (if exists) of their knowledge units in relation α_L.

d) Any test should have a presentation as its predecessor and any presentation should be followed by a test.

The following example should explain the definition:

Example 1. Let $L = \langle k, k', k'' \rangle$, $R_k = \{p_1, p_2\}$, $R_{k'} = \{p_{1'}, p_{2'}\}$, $R_{k''} = \{p_{1''}, p_{2''}\}$, $T_k = \{t_1, t_2\}$, $T_{k'} = \{t_{1'}\}$, $T_{k''} = \{t_{1''}, t_{2''}\ t_{3''},\}$ and $\alpha_L = \{(k', k'')\}$. Some potential scenarios for course L are:

$$s_1 = \langle p_{1'}, t_{1'}, p_{2''}, t_{1''}, p_1, t_2 \rangle,$$
$$s_2 = \langle p_2, t_2, p_{2'}, t_{1'}, p_{1''}, t_{1''} \rangle,$$
$$s_3 = \langle p_2, t_1, p_{1'}, t_{1'}, p_{1''}, t_{2''} \rangle.$$

Let S_L be the set of all scenarios for course L. Notice that the cardinality of set S_L is dependent on the number of knowledge pieces occurring in L, the numbers of presentations of each knowledge piece and the order α_L. Generally, this number may be very large, but always finite.

General conception of learning process relies on the assumption that similar students should learn in the same way, namely by the same learning scenario. So, when a learner takes up a course the system creates his initial profile on the base of student's characteristics. In following step this profile is used to classify the student to a set of similar learners. This isolated set of learner's serves to determine an opening learning scenario. At the beginning of system functioning, opening learning scenario is assigned to a given class by experts or pedagogical agent. In the situation when there is a class of students that are similar to a new learner and that have finished chosen course with success, opening scenario is determined by the analysis of their final scenarios.

When learning procedure starts it offers to the student first sequence of hypermedia pages indicated by the scenario for first piece of knowledge and the student starts to learn. Next the student has to pass a test suitable to the knowledge he has just learnt. According to the test results a pedagogical agent decides if the student should continue learning in keeping with earlier steady learning scenario. If test results are not sufficient the agent can change scenario by choosing, from presentations available for a given knowledge unit, this one that has not been used yet and better suits to actual learner characteristics. Sometimes the agent may change the order of presented knowledge units. In every case the actions of pedagogical agent aim at delimiting the more effective way of learning for a given student.

All decisions that have been made during the whole course are remembered and used to modify student's profile and to create final learning scenario. In consequence preliminary learners' classification could be no more correct. Therefore, at the end of learning process the student may be reclassified. Final scenario that have brought the student to success together with final scenarios of the other members of the class will be used to determine opening learning scenario for a new student classified by the system into the same category.

3 Learner Profile

A learner profile is student's representation in education system. It should include the entire information concerning the student and important from system adaptability point of view. This information can be divided in two main categories: personal features and user – system interaction history.

First of them would contain the data that permit student's identification and those, which are connected with learner's education level, i.e. learner's identifier, name and forename, birthday, age, completed education (primary, secondary, graduate and post graduate). For the attribute age the following values have been assigned: young, middle and advanced corresponding to the (18-35⟩, (35-65⟩, (65 and more) age ranges.

Personal features comprise also Intelligence Quotation (IQ) with values: low (less then average), middle (average) and high (above average) and individual student's learning style. The latest attribute will be represented by model developed by R. Felder and L. Silverman [5]. The model considers learner's behavior on four bipolar dimensions: perception (sensitive, intuitive), receiving (visual, verbal), processing (active, reflective) and understanding (sequential, global). To assess student's preference along the particular model dimension Index of Learning Styles questionnaire developed by R. Felder and B. Soloman [6] should be applied. The results obtained are presented as a pair, where first element refers to learner's preferred direction for every dimension, and the second is a score on a scale 1-11 that indicates the intensity of student's behavior in a given direction. This information will be useful for pedagogical agent to determine alternative for a given student way of knowledge unit presentation while previously used led to failure.

Definition 2. By individual learning style *ILS* we call Cartesian product:
$ILS = PER \times REC \times PRO \times UND$, where:
$PER=\{SEN,INT\}$, $REC=\{VIS,VER\}$, $PRO=\{ACT,REF\}$, $UND=\{SEQ,GLO\}$,
and $SEN = (sensitive,\ i)$, $INT = (intuitive,\ j)$, $VIS = (visual,\ i)$, $VER= (verbal,\ j)$, $ACT = (active,\ i)$, $REF = (reflective,\ j)$, $SEQ = (sequential,\ i)$, $GLO = (global,\ j)$,
where $i, j = 1, 2, \ldots, 11$ and if $i>0$, then $j = 0$ and else, if j>0, then $i = 0$ for: *SEN* and *INT, VIS* and *VER, ACT and REF, SEQ* and *GLO.*

The second category of learner profile data cover dynamic information gathered by system during learner – system interaction. They comprise the information concerning completed courses, i.e. course identifier, opening and final learning scenarios, as well as currently realized course, i.e. course identifier and opening scenario. All learning scenarios remembered in student's profile have the same structure consistent with the definition 1. For finished and currently realized courses all the tests scores in opening scenarios are equal to 0. In final scenarios there will be the real scores obtained by the student in successfully passed tests following related knowledge units presentations.

Let's notice that in looking for the similarity between students not all of the attributes would be of the same importance. So, it was decided that there will be taken into account only these attributes, which are supposed to significantly influence the effects of learning processes. They are the basis of classification, hence will be called *basic attributes*. To the set of *basic attributes* – ***B*** will be included (with regard to legibility the notations will refer to first letters of attributes names and values):

- age – $AG = \{YO,MI,AD\}$,
- sex – $SX = \{MA,FE\}$
- intelligence quotient – $IQ = \{LQ,MQ,HQ\}$,
- education – $ED = \{EE,SE,HE,PE\}$,
- individual style of learning – *PER*, *REC*, *PRO* and *UND*.

Remaining student's data containing:

- learner identifier – *PI*,
- name and forename – *NF*,
- date of birth – *BD*,
- finished courses (opening and final learning scenario) – $LF = \{(I_1,Os_1,Fs_1),...,(I_k,Os_k,Fs_k)\}$
- currently realized course (opening learning scenario) – $LA = (I_a,Os_a)$,
- score on the scale 1-11 for each attribute: *SEN, INT, VIS, VER, ACT, REF, SEQ* and *GLO*

are included to the set of *supplementary attributes* – ***A*** and in a given moment does not participate in student's classification.

4 The Classification

The basic attributes define the characteristics of a class and consequently they determine the learners' profiles distribution into classes. The remaining (supplementary) attributes reflecting the data about students could be used to the actualisation of classification or to the future investigations.

Definition 3. Let $\boldsymbol{B} = \{B_1,...,B_N\}$. By profile classification $\boldsymbol{C}$ we call Cartesian product $\boldsymbol{C}=B_1\times...\times B_N$. Each element $C=(b_1,...,b_N)$ of product $\boldsymbol{C}$ we call a class of classification. In the other words, $C\in \boldsymbol{C}$, if and only if $b_1\in B_1,...,b_N \in B_N$.

Definition 4. We say that value b of the attribute B represents a class $C = (b_1,...,b_N)$ if and only if b belongs to set $b_1,...,b_N$.

Definition 5. We say that learning scenario s belongs to a class $C = (b_1,...,b_N)$, if and only if there exists at least one profile p that belongs to C, for which s is final scenario.

Example 2. Let $\boldsymbol{C} = AG\times SX\times IQ\times ED\times PER\times REC\times PRO\times UND$, where *AG, SX, IQ, ED, PER, REC, PRO, UND* are *basic attributes* with values defined in previous section. To the category $C_j = (YO,FE,HQ,SE,INT,VIS,ACT,SEQ)$, will be included the profiles of students who:

- are young (*YO*),
- are female (*FE*),

- have high intelligence quotient (*HQ*),
- have completed secondary education level (*SE*),
- like innovations and dislike routine calculations (*INT*),
- prefer knowledge presented in graphical form (*VIS*),
- like active experimentation (*ACT*),
- prefer material presented in steady progression of complexity and difficulty (*SEQ*).

In the classification mentioned above as *basic attributes* were assumed these attributes and their values that have been supposed to influence the effects of learning. However, it cannot be assumed that the selection of attributes and their values is entirely consistent with the postulate of dependency between learning scenarios and basic attributes. Scenario verification and connected with it verification of classification takes place during classification process.

When the student is finishing the course without changing of the opening scenario, he stays in previously determined class. If the scenario is changed, then the pedagogical agent decides to transfer the student to another class. After some period of time the data gathered by system, especially opening and final learning scenarios of completed courses, will by analyzed in detail to modify the classification and to determine new opening scenario for each class and every course of actualized classification.

5 Distance Function between Scenarios

It is possible to calculate the distance between scenarios. To measure the distance the following constraints should be taken into account:

- The distance of 2 ordered sets of knowledge pieces' presentations appearing in these scenarios,
- The distance of 2 ordered sets of tests.

By d: $S_L \times S_L \rightarrow [0,1]$ we call the distance function between scenarios, where [0,1] is the set of real numbers, not smaller than 0 and not larger than 1.

In a scenario $s \in S_L$ we distinguish the following sequences: sequence s_P consisting of knowledge presentations occurring in s, and sequence s_T consisting of tests occurring in s. The following example should illustrate the notions.

Example 3. Let $L = <k_1, k_2, k_3, k_4>$ and $T = <t_1, t_2, t_3, t_4>$. If $s = <p_1, t_1, p_2, t_4, p_3, t_2, p_4, t_3>$ where presentation p_i corresponds to knowledge piece k_i for i=1,…,4, then $s_P = <p_1, p_2, p_3, p_4>$ and $s_T = <t_1, t_4, t_2, t_3>$.

The idea of the distance between 2 scenarios s and s' is the following: Firstly, we measure the distance $d_1(s_P,s_P')$ between sequences s_P and s'_P, next the distance $d_2(s_T,s_T')$ between sequences s_T and s'_T is determined. Finally, the distance $d(s,s')$ is calculated as:

$$d(s,s') = \frac{1}{1+\beta}(d_1(s_P,s_P') + \beta \cdot d_2(s_T,s_T')),$$

where β (0≤β≤1) is the parameter representing the participation degree of the distance between tests in the distance d.

In determination of distance $d_1(s_P,s_P')$ 2 constraints should be taken into account: the first is the distance between 2 orders of knowledge pieces presented in scenarios s

and *s*', and the second refers to the distances between the presentations of each knowledge piece. These constraints should be calculated for each knowledge piece as follows: For given knowledge piece *k* let *r* and *r*' be its corresponding presentations with indexes *i* and *i*' in scenarios *s* and *s*' correspondingly. The participation of *k* in distance $d_1(s_P,s_P')$ is equal to $|i-i'|$ if *r*=*r*' or $|i-i'|\cdot\gamma$ if *r*≠*r*', where γ ($1<\gamma\le 2$) is the parameter which causes the increase of the distance between knowledge pieces in distance $d_1(s_P,s_P')$ in case when their presentations are different. Such defined distance $d_1(s_P,s_P')$ is consistent with one of distances between linear orders proposed among other in Arrow's pioneer work of choice theory [1]. In this work we need it to be normalized by dividing by number $\gamma n(n+1)/2$.

The distance $d_2(s_T,s_T')$ between 2 test orders may be calculated in a similar way as for distance $d_1(s_P,s_P')$. For given knowledge piece *k* let *t* and *t*' be its corresponding tests with indexes *i* and *i*' in scenarios *s* and *s*' respectively. The participation of *k* in distance $d_2(s_T,s_T')$ is equal to $|i-i'|$ if *t*=*t*' or $|i-i'|\cdot\eta$ if *t*≠*t*', where η ($1<\eta\le 2$) is the parameter which causes the increase of the distance between knowledge pieces in distance $d_2(s_T,s_T')$ in case when their corresponding tests are different. Next $d_2(s_T,s_T')$ should normalized by dividing by number $\eta n(n+1)/2$.

Example 4. Assuming that $\beta=0.5$, $\gamma=2$ and $\eta=1.5$, let scenarios s_1 and s_2 be defined as in Example 3. The distances d_1 and d_2 for scenarios s_1 and s_2 are equal 2/3 and 1/1.5 correspondingly. Thus the distance $d(s_1,s_2)$ is equal $(2/3+0.5\cdot(1/1.5))/2 = 0.5$.

It is not difficult to prove that such defined distance function is a metric.

6 Determining the Opening Scenario Using Consensus Methods

To determine the opening scenario that is proposed to the learner the system should use the information it possesses in its database. In this database there are stored the profiles, including among others the final scenarios, of these learners who have been classified into the same class as our new student and have finished (with positive results) the same course, which the new student wants to learn. Although these learners have similar features, the scenarios, on the basis of which they have finished the course successfully, may be different. However, we assume that these scenarios are an essential resource of information for defining the opening scenario for the new learner. Because of their differences we propose to determine the opening scenario as the consensus of existing scenarios.

For given course *L* and a set of scenarios $s_1,\ldots,s_n$ for this course, we define their consensus as follows:

Definition 3. By the consensus of scenarios $s_1,\ldots,s_n$ we call a scenario $s^*\in S_L$ for which the following condition is satisfied:

$$\sum_{i=1}^{n} d(s^*,s_i) = \min_{s\in S_L} \sum_{i=1}^{n} d(s,s_i).$$

Notice that there have been worked out many criteria for consensus choice [13], but the criterion presented in Definition 3 is most often used in practice because it guarantees that the consensus at best represents the given data versions which are inconsistent.

It is also worth to note that the number of elements of set S_L may be very large and this could make the consensus choice task to be a complex one. Besides it has been proved [2] that the problem of determining an order that generates the smallest sum of distances to given orders (as a particular case of our task) is a NP-hard one. For these reasons below we propose a heuristic algorithm for this task.

The idea of this algorithm is the following: For each scenario s from scenarios $s_1,\ldots, s_n$ its presentations' order s_P is calculated. There will be worked out 2 procedures. The first of them determines the consensus s_P^* for orders s_P and the second determines the consensus s^* by selection of corresponding tests to presentations occurring in order s_P^*. These procedures a presented below:

Procedure 1. Determining the consensus s_P^* for orders $s_{1P},\ldots, s_{nP}$.

```
BEGIN
1. For each $i=1,\ldots,n$ calculate $S(s_{iP}) = \sum_{j=1}^{n} d_1(s_{iP}, s_{jP})$.
2. Select such $i$ that sum $S(s_{iP})$ is minimal.
3. Let $s_P^* := s_{iP} = <p_1,\ldots, p_n>$.
4. For $i=1,\ldots,n$ do
        For $j= i+1,\ldots,n$ do
        Begin
        4.1. If pair $<k_i,k_j>$ of knowledge pieces to which are corresponding $p_i$ and $p_j$ does not appear in order $\alpha_L$ then create $s'_P$ from $s_P^*$ by reordering $p_i$ and $p_j$.
        4.2. If $S(s'_P)<S(s_P^*)$ then $s_P^* := s'_P$.
        End.
END.
```

Procedure 2. Determining the consensus s^* for scenarios $s_1,\ldots, s_n$.

```
BEGIN
1. Calculate consensus $s_P^*$ using Procedure 1. Let $s_P^* = <p_1,\ldots, p_n>$.
2. Create a scenario $s = <p_1, t_1,\ldots, p_n, t_n>$ where $t_i$ are tests and pair $p_i$, $t_i$ is corresponding to a knowledge piece $k_i$ in course $L$.
3. Calculate $V(s) = \sum_{i=1}^{n} d(s, s_i)$.
4. For $i:=1$ to $n$ do:
        Choose such test $t$ from $T_{k_i}$ and replace the corresponding test to $k_i$ in $s$ so that value $V(s)$ is minimal.
5. Let $s^* := s$.
END.
```

It is possible to prove that the computational complexity of Procedure 1 is $O(n^2)$ and of Procedure 2 is $O(mn)$ where m is the maximal number of tests corresponding to knowledge pieces from L. Thus the complexity of the algorithm is $O(n^2+ mn)$.

7 Implementation

For the implementation of an experimental e-Learning environment it is usually impossible to apply only standard authoring tools for web-based systems development such as WebCBT or Toolbook. We must consider less standardized implementation that enables full control over most important aspects of the implemented e-Learning environment.

Most modern web-based systems are implemented using 3-layer architecture, where presentation, application and database layers could be distinguished [15]. In this type of architecture we are considerably free with selection of the particular tool for the implementation of each layer. To implement the presentation layer we can use: HTML or XHTML also with some dynamic elements such as JavaScript or Java applets. More flexible solution is offered by using XML for information representation and XSL for its presentation. To implement applications rich with multimedia elements, Macromedia Flash seems to be very good solution. There are however many other potential solutions, such as W3C SVG (Scalable Vector Graphics) – a language for describing 2D vector graphic in XML or QuickTime. Nowadays it is considered to use the 3D interfaces not only in computer games but also in education applications. To implement interfaces of this kind we can apply VRML, CULT 3D or Adobe Atmosphere. Usually to present such applications, the web browser must use special plug-ins.

In web-based systems presentation layer is very often closely integrated with application layer. This is so, because in application layer scripts or components generate HTML code, which is sent and only interpreted by a client. Of course JavaScript, Flash or Java applets require from client computer not only the ability to interpret simple HTML code but also to run parts of application code.

Database layer should be built using such a database management system, which is capable not only to store data, but also to provide flexible data access. It should provide executing SQL queries and retrieving results as relational record set and XML documents. This solution allows designers to build applications for different target users. It can be used by applications running in local networks, in the Internet as well as through wireless equipment like mobile phones and handholds.

8 Conclusions

In this paper a conception for a model of intelligent e-Learning environment is presented. We propose to use learner profiles to classify students into classes, which serve to group the students with similar features. Such classification is useful because owing to it one can determine a sensitive scenario for a new learner. This scenario is calculated by consensus methods. The final profile and scenario of the student after his successfully finishing the course will be stored and the system may use this information for other new students. Future works should concern realization of this conception in a computer program, which can function in Web environment.

References

1. Arrow, K.J.: Social Choice and Individual Values. Wiley New York (1963),
2. Barthelemy, J.P., Janowitz, M.F.: A Formal Theory of Consensus. SIAM J. Discr. Math. 4 (1991) 305-322,
3. Boyle T., Design for multimedia learning, Prentice Hall Europe (1997),
4. Chandnani K.: International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet. SSGRR (2003), http://www.ssgrr.it/en/ssgrr2003w/papers/163.pdf,
5. Felder R, Silverman L.: Learning and Teaching Styles in Engineering Education, ASEE J. of Engng. Education, 78 (7) (1988) 674-681,
6. Felder R., Soloman B.: Index of Learning Styles (1998), http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS-a.htm,
7. Garro A., Palopoli L.: An XML Multi-agent System for E-learning and Skill Management. Agent Technologies, Infrastructures, Tools, and Applications for E-Services (2002) 283-294,
8. Kukla E., Nguyen N.T., Sobecki J.: The consensus based tutoring strategy selection in CAL systems, World Transactions on Engineering and Technology Education, vol. 1, Melbourne (2001) 44-49,
9. Kukla E.: Outline of tutoring strategy construction method for multimedia intelligent tutoring systems, Multimedia and Network Information Systems Conference – MISSI'2002, Wrocław (2002) (in Polish),
10. Kukla E., Nguyen N.T., Sobecki J., Daniłowicz C., Lenar M.: A Model Conception for Learner Profile Construction and Determination of Optimal Scenario in Intelligent Learning Systems, Lecture Notes in Artificial Intelligence, vol. 2774, Springer-Verlag, Berlin Heidelberg New York (2003) 1216-1222,
11. Lance G.N. Williams W.T.: A General Theory of Classificatory Sorting Strategies. 1. Hierarchical systems. Computer Journal, Vol. 9 (1966) 373-380,
12. Nguyen, N.T.: Consensus Systems for Conflict Solving in Distributed Systems. Journal of Information Sciences vol. 147 (2002) 91-122,
13. Nguyen, N.T.: Using Distance Functions to Solve Representation Choice Problems. Fundaments Informaticae vol. 48 (2001) 295-314,
14. Rasmussen E.: Clustering algorithms. In: Information Retrieval: Data Structures & Algorithms, Prentice Hall, Englewood Cliffs (1992) 419-442.
15. Sobecki J., Morel K., Bednarczuk T.: Web-based Intelligent Tutoring System with Strategy Selection Using Consensus Methods. Springer-Verlag Series on Advances in Soft Computing, (2003).

An Intelligent GIS-Based Spatial Zoning System with Multiobjective Hybrid Metaheuristic Method

Bong Chin Wie and Wang Yin Chai

Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak,
Jalan Dato' Musa, 93400 Kota Samarahan, Sarawak, Malaysia
(cwbong, ycwang)@fit.unimas.my

Abstract. This paper presents a multiobjective hybrid metaheuristic approach for an intelligent spatial zoning model in order to draw territory line for geographical or spatial zone for the purpose of space control. The model employs a Geographic Information System (GIS) and uses multiobjective combinatorial optimization techniques as its components. The proposed hybrid metaheuristic consists of the symbiosis between tabu search and scatter search method and it is used heuristically to generate non-dominated alternatives. The approach works with a set of current solution, which through manipulation of weights are optimized towards the non-dominated frontier while at the same time, seek to disperse over the frontier by a strategic oscillation concept. The general procedure and its algorithms are given as well as its implementation in the GIS environment. The computation has resulted in tremendous improvements in spatial zoning.

1 Introduction

In this paper we report an application of Geographic Information System (GIS) and multiobjective hybrid metaheuristic algorithms to the development of a decision-support system used in the Spatial Zoning Procedure (SZP). Spatial zoning is defined as to draw lines for a boundary [1]. It is to partition geographical zones with territory, subject to some side constraints [2]. A combinatorial optimization model is formulated to represent the SZP problem and a multiobjective metaheuristic solution procedure is developed to solve the optimization model. The SZP is tremendous important that it reacts on how we take control on the space of a particular region [3]. The process is not only relevant to election system, but also important in school districting, business territory, law enforcement and forests planning related system.

SZP problem has been implicitly characterized as a combinatorial optimization problem from 1961 to obtain the optimal arrangement of a group of discrete entities in a way that additional requirements and constraints are satisfied [4, 5, 6]. After 1995, the solution of SZP has been shifted from the exact to heuristic method and then the latest trend to a metaheuristic method. Basic algorithms in exact method were first produced in 20 years ago [7]. However, computing exact optimal solutions for spatial zoning problem is computationally intractable because it is one of the numerous NP-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 769-778, 2004.

hard combinatorial optimization problems [8]. In contrast, the earliest heuristic method was a mild steepest descent heuristic. Nevertheless, it works well only on small problems. Later, there are other heuristic procedures developed to solve the problem such as simulated annealing, tabu search, genetic algorithm, GRASP, and other hybrid methods [9]. In addition to heuristic methods, there is a recent trend of metaheuristics, which combine such heuristic tools in more sophisticated frameworks [8, 12]. Despite the recent progress made by metaheuristics, there are still many instances in these methods become trapped in poor local minima from which they are unable to escape [10].

The rest of this paper is organized as follow. In Section 2, the Multiobjective Hybrid Metaheuristic Procedure for the SZP model is described. This is followed by implementation and application in Section 3 and by conclusion in Section 4.

2 The Multiobjective Hybrid Metaheuristic Procedure

In this study, the hybrid metaheuristic composes of the tabu search and scatter search concept to improve the search process. Symbiosis of these existing metaheuristic methods helps support this spatial data processing with intensification and diversification process. We include adaptive memory structure to identify wider range of alternatives and to analysis the relationship between alternatives with multiobjective problem definition and the measurement or approximation to non-dominated front.

In the following, we discuss and present the components and a step-by-step description of the metaheuristic solution procedure from the basic understanding and definition, initialization, neighboring move, dominancy comparison, quality measurement and adaptive memory structure used. In last section, we also present the overall algorithm for the metaheuristic designed for the multiobjective SZP.

2.1 Basic Understanding and Definition

In term of SZP, a multiobjective decision making requires that means-end relationships be specified, since they deal explicitly with the relationship of attributes (non-geographical) of alternatives to higher-level objectives of the decision makers. Therefore, the role of the multiobjective approaches is to provide a framework for designing a set of alternatives. Each alternative is defined implicitly in term of the decision variables and evaluated by means of objective functions. The approach derives attributes of alternatives from the preferences among objectives and the functions relating attributes to objectives. An attribute here is a concrete descriptive variables meanwhile an objective is a more abstract variable with a specification of the relative desirability of the levels of that variable. Consequently, we will store the input data in GIS in the form of map layers. Each map layer contains a set of objects so that we can process them and derive the alternatives to produce final zoning plans as the outcomes by defining the relationship between the objectives and the underlying attributes of the objectives contained in geographical space. It is important to note that the process typically requires an algorithm specifically designed to tackle SZP. Usually, it is not possible to use the standard set of operations available in GISs to generate the spatial multiobjective alternatives because the spatial multiobjective analysis goes beyond the standard GIS tools.

Among most of the spatial zoning applications, there are several important objectives. In our case, we use to minimize all the objective functions. Firstly, to qualify a solution, an important criterion is that the zones should be compact which the idea is to prevent the formation of odd-shaped zones. Another commonly accepted objective is with respect of the average of resources capacity such as population equality, a must criterion for political districting. Therefore, we define two compulsory objectives for a general multiobjective spatial zoning problem: minimization travel distances (Compactness), and minimization of the average deviation of resources capacity or size. However, in some particular spatial zoning application, there are application-dependent objectives. For instance, personal income will be a socio-economic homogeneity to ensure a better representation of residents, which share common concerns for a political districting.

Notation and Scaling:
Multiobjective SZP Problem in this study includes a set of n parameters (decision variables), a set of k objective functions, and a set of m constraints. Objective functions and constraints are functions of the decision variables. The optimization goal is to minimize $y = f(x) = (f_1(x), f_2(x),..., f_k(x))$, subject to $e(x) = (e_1(x), e_2(x),..., e_m(x)) \leq 0$ where $y = (x_1, x_2,..., x_n) \in X$ and $y = (y_1, y_2,..., y_k) \in Y$ and x is the decision vector, y is the objective vector, X is denoted as the decision space, and Y is called the objective space. The constraints $e(x) \leq 0$ determine the set of feasible solutions. Throughout this paper, objective indices are written in superscript. Let us also define dominance and superiority:

- The point z_1 *dominates* the point z_2 if and only if $z_1 \geq z_2$ and $z_1 \neq z_2$ (i.e. if $z_1^k \geq z_2^k$ for all objectives k and $z_1^k > z_2^k$ for at least one objective k). The point z_1 is *dominated* by the point z_2, if the point z_2 dominates the point z_1.
- Solution x_1 is *superior* to solution x_2 if the point $f(x_1)$ dominates the point $f(x_2)$. Solution x_1 is *inferior* to solution x_2 if the point $f(x_2)$ dominates the point $f(x_1)$. If the point $f(x_1)$ is non-dominated, then x_1 is *non-inferior.*

The set of all non-inferior solutions are referred to as the Pareto optimal set or the efficient set. The set of all non-dominated points is referred to as the non-dominated set. An efficient solution for Multiobjective SZP should be Pareto optimal, and the solutions are uniformly sampled from the Pareto optimal set.

2.2 Diversified Seed Solution Initiator

A wide exploration of the solution space is important to effectively navigate the algorithm into various regions of the search domain. Therefore, the first step in the proposed approach is to create an initial solution with diversification purpose to encourage the search process to examine random regions of the solution space. Given a set I of basic units and attributes in a layer form, we will generate a set of initial solution with an algorithm of seed solution generator with random diversification. Firstly, a seed unit is selected randomly to initialize a zone. Then, this zone is extended gradually by adjoining to one of its adjacent units. The zone is complete whenever no adjacent units are available or when its capacity attains *P*. When there is left over basic units or there is any basic unit that has not been assigned with a zone number, they will be merged with least populated zone. If all zones in the final zoning plan are continuous, it becomes initial seed solution. In other words, at the end of this process, the initial solution x_0 is a zoning plan that made up of *m* continuous zones, some of which may be infeasible with respect to some of the objectives defined.

2.3 Neighboring Move

After we obtain the initial solution (a zoning plan), the proposed algorithm conducts a neighboring tabu move strategies to prepare for generating an intensified solution to return to attractive regions of the solution space to search them more thoroughly. In this case, we adopts the move strategies used by [2] with two neighborhoods, $N_1(x)$ and $N_2(x)$. The first $N_1(x)$ is made up of all solutions reachable from *x* by moving a basic unit *i* from its current zone *j* to a neighbor zone *l* without creating a non-contiguous solution. Such a move is said to be of Type I and denoted by (*i, j, l*). The second neighborhood $N_2(x)$ is made up of all solutions that can be reached from *x* by swapping two border units *i* and *k* between their respective zones *j* and *l*, again without creating discontinuities as shown in Fig. 1. Such a move is said to be of Type II and denoted by (*i, k, j, l*). On the other hand, to help prevent cycling, whenever a move (*i, j, l*) or (*i, k, j, l*) is performed, any move that puts *i* back into *j* or unit *k* back into zone *l* is declared as tabu.

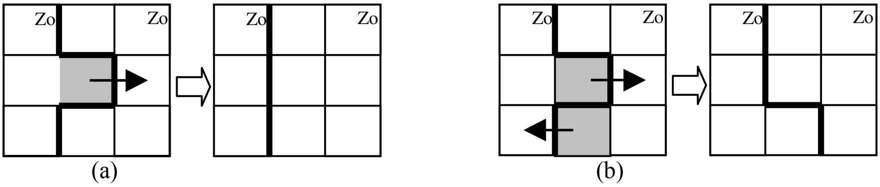


Fig. 1. Neighboring move strategies (a) Type I– a basic unit, *i* is moved to neighboring zone *l* from current zone *j*. (b) Type II – two basic units, *i* and *k* are swapped between their respective zones *j* and *l* (Bold line indicate a territory)

Our philosophy on strategic oscillation of the Scatter Search (SS) in Seed Solution Initiator and Neighboring Tabu Move helps provide the necessary mixture of intensification and diversification. By moving until hitting a feasible boundary, where normally a heuristic method would stop, we extend and evaluate the neighborhood definition for selecting modified moves, to permit crossing of the boundary. These strategies are used respectively to focus the search into more promising regions, modifying choice rules to stimulate combinations of good moves and improved solutions. It leads the search to unexplored regions of the solution space and generates a neighboring subset. We then combine the subset with spatial adjoining or merging process.

In other words, the neighboring move provides a pool of basic units to generate neighboring subsets to construct solutions by combining various elements with the aim that the solution based on the combined elements will exploit features not contained separately in the original elements. The neighboring subsets contain dynamic number of basic units depending to the adjacent units and this helps to reduce the computation time in the procedure. Therefore, the neighboring subset generator and combination work with the underlying concept of SS are indeed similar to evolutionary methods but have intimate association with tabu search metaheuristic. Unlike genetic algorithms, they operate on a small set of solutions and employ diversification strategies of the form proposed in tabu search with limited recourse to randomization. The solution combination method transforms the given subset of solutions produced by the dynamic neighboring subset into a combined solution. The combination is analogous to the crossover operator in the genetic algorithm but it is capable of handling the spatial data features.

2.4 Dominancy Comparison to Measure Manhattan Distance Norm and Set Distance Proximity Function

In this step, we determine the weight vector (λ) for each of the objective functions vector or point. We use weight vector from the λ-vector space Λ for each of the objective functions that defined as

$$\Lambda = \{\lambda \in \Re^{j} \mid \lambda^{k} \in [0;1] \wedge \sum_{j=1}^{j} \lambda^{j} = 1\} \tag{1}$$

The neighboring move will first insure optimization towards the non-dominated frontier. Then, we will to set and modify the weights which are obtained from the decision maker (DM), so that the point moves away from the other points, ideally having the points equidistantly spread over the frontier. Therefore, each element in the weight vector is set according to the proximity of other points for that objective. However, we only compare a point with the points of the current solution to which it is non-dominated. The closer another point is, the more it should influence the weight vector. The closeness is measured by a distance function (*d*) based on some metric in the objective function space and the range equalization weights. The influence is given by a decreasing, positive value of proximity function (*g*) on the distance. In practice, the proximity function *g(d)=1/d* has shown to work well, and the Manhattan distance norm as following:

$$d(z_i^{k}, z_j^{k}, \pi) = \sum \pi^{k} \mid z_i^{k} - z_j^{k} \mid \tag{2}$$

The distance norm, π used on the objectives is then scaled by the range equalization factors. This factor is used to equalize the ranges of the objectives, and calculated as

$$\pi_j = \frac{1}{R_j}, j = 1,\ldots,J \tag{3}$$

where R_j is the (approximate) range of objective *j* given a set of points. Objective function values multiplied by range equalization factors are called normalized objective function values.

2.5 Measurement with Multiobjective Tchebycheff Scalarization Function

The use of a multiobjective acceptance rules in the quality measurement is crucial in approximating the non-dominated solution for the multiobjective spatial zoning problem. Therefore, we use the quality dimension in tabu search that usually refers to the ability to differentiate the merit of solutions visited during the search. In this context, we use memory to identify elements that are common to good solutions or to paths that lead to such solutions. Operationally, quality becomes a foundation for incentive-based learning, where inducements are provided to reinforce actions that lead to good solutions. The flexibility of the memory structures allows the search to be guided in a multiobjective environment, where the goodness of a particular search direction may be determined by more than one function. In this study, we concentrate the quality counter with achievement scalarizing function, which takes into account on the weight vector, reference set of the optimal solution and also each objective function scaling. The advantage of scalarizing function is the possibility of forcing particular solutions to explore the desired regions of non-dominated set. It allows for problem specific heuristics for construction of initiate solution and/or local improvement. Accepting a randomly generated solution from the neighborhood may modify each generating solution from the neighboring move.

Then, for the purpose for performance comparison, we use Achievement Tchebycheff Scalarizing Function (ATSF) as (5) to qualify the generated solution with the reference point, at the objective functions *f*(*x*) or *z* where z^0 is a reference point, $\Lambda = [\lambda_1,...,\lambda_j]$ is weight vector, ρ is sufficiently small positive number. The use of ATSF will be good at locating non-supported non-dominated points [11].

$$S(z,z^0,\Lambda,\rho) = \max_j\{\lambda_j(z_j{}^0 - z_j)\} + \rho\sum_{j=1}^{j}\lambda_j(z_j{}^0 - z_j) \tag{4}$$

2.6 Adaptive Memory Procedure

Last but not least, we embed the proposed method within an Adaptive Memory Procedure (AMP). The AMP is based on the idea that the high quality solution are stored and used to construct other high quality solution. In our problem, the solutions are the set of zoning plans. The method therefore stores the constantly updated set of potentially pareto-optimal solutions in a Reference Set (*RS*). *RS* is empty at the beginning of the method. We will continuously update it whenever a new solution is generated. Updating the set of potentially Pareto-optimal solutions with solution *x* consists of adding *z* to *RS* if no point in *RS* dominates *z*, and removing from *RS* if all points dominated by *z*. The process of updating *RS* may be very time consuming. Thus, we use the following rules to reduce the computational requirements:

- New solution *y* obtained from *x* should be used to update *RS* only if it is not dominated by *x*.
- New potentially Pareto-optimal solutions could be added to *RS* only if they differ enough from all solutions contained already in this set. We propose to use a threshold defining the minimum Euclidean distance in the space of normalized

objectives between solutions in *RS*. A new potentially Pareto-optimal solution is neglected if it is closer to at least one solution in *RS* than the threshold.

We have observed experimentally that the solutions added to *RS* in early iterations have a good chance to be removed from this set in further iterations. In other words the time spent on updating *RS* in early iterations is likely to be lost. Thus, it is possible to neglect updating *RS* in a number of starting iterations.

Whenever a new solution is created, the zoning plan becomes the member of the memory. The size of the memory is kept constant that its worst elements are regularly replaced by better ones. A data structure called quad trees is used to accelerate the process of updating *RS*. The quad tree of points where no point dominates any other point is applied and it is computationally very efficient methods to determine whether points in the tree dominate a given new point and to retrieve point in the tree, which it dominates.

2.7 Summary of the Algorithm

The step-by-step description of the high-level algorithm is provided as following:

```
Begin
SeedSolutionGenerator()
Set $RS=\varnothing$ and $\pi=1/n$ for all objectives j
Repeat for each boundary
  NeighboringTabuMOVE()
  For each neighbour, x
    Get $\lambda$ from DM
    For each x' where $f(x)\neq f(x')$
      Get f(x') for all objectives
      If x' where f(x') is non-dominated by f(x)
        Measure the Manhattan distance norm
        Set distance proximity function
        For all objective j where $f_j(x) < f_j(x)$,  Set $\lambda_j = \lambda_j + \pi_j w$
      End if
    Loop
  Loop
  Normalise ($\lambda$)
  Calculate ATSF(or WSF)
  Select solution y with minimum ATSF (or WSF) and spatial continuity checking
  SolutionCombination()
  If y is non-dominated by any objective function in RS, then Update RS with y and $\pi_j$
Until for each boundary
End
```

Fig. 2. Procedure for Multiobjective Hybrid Metaheuristic for the Spatial Zoning Model

3 Implementation

The algorithm just described was implemented in ArcGIS version 8.1, a GIS software products of the Environmental Software Research Institute (ESRI). ArcGIS, a high-end GIS software, is ESRI's flagship product, which has the capabilities of automation, modification, management, analysis and display of geographical information. The implementation coding was conducted with Visual Basic Application (VBA) embedded in ArcGIS and run on a Pentium IV 2.4 GHz PC with 256MB RAM. The input data to the multiobjective spatial zoning problems was stored in the form of map layers called Shapefiles format, which were handled and visualized using the ArcGIS. Each map layer contained a set of objects that were considered as elements of alternative solutions. The map layers were first processed to define the relationship between the zoning objectives as mentioned earlier and the underlying attributes of the objectives contained in geographical space. Some of the basic operations used included operations on spatial relationship of connectivity, continuity, proximity and the overlay methods. The algorithm just described was coded and test on a political districting. The political districting data used in the model is as following:
I = The set of all basic units which are represented by enumeration areas (EAs). For each unit, all population data and geographical data are available.
J = The set of basic units used as potential ' 'seeds' ' .
m = The number of zones to be created is given
p_i = The population capacity of unit i
$[a, b]$ = interval of the population capacity of any zone must lie.

Three objectives are considered in this model. Each of the function below corresponds to one of the goals.

Goal 1: The objective function f_1 corresponding to the goal 1 measures the average deviation of the population. Indeed, population equality, $P_j(x)$ is the population of district j. The population of each district is lie within some interval $[a,b]$ = $[(1-\beta)\overline{P},(1+\beta)\overline{P}]$ where $0 \le \beta < 1$. The objective is formulated in the following way:

$$f_1 = \frac{\sum_{j \in J} \max\{P_j(x) - (1+\beta)\overline{P}, (1-\beta)\overline{P} - P_j(x), 0\}}{\overline{P}} \tag{5}$$

Goal 2: The objective function f_2 corresponding to the goal 2 measures compactness by measuring the total length of all boundary lengths between zones, excluding the outside boundary of the territory:

$$f_2 = \sum_{j \in J} (1 - \frac{2\pi\sqrt{A_j(x)/\pi}}{R_j(x)}) \Big/ m \tag{6}$$

where $R_j(x)$ and $A_j(x)$ are the perimeter and area of j in the solution x.

Goal 3: The objective function f_3 corresponding to the goal 3 measures a socio-economic homogeneity, S. the reasonable objective is to minimize the sum over all zones j, of the standard deviation $S_j(x)$ of by the average income of each basic unit in the zone.

$$f_3 = \sum_{j \in J} S_j(x) / \overline{S} \tag{7}$$

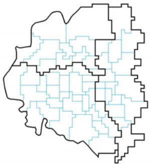
3.1 Experiment

Since it is unrealistic to test all combination of candidate values for every aspect of the proposed multiobjective hybrid metaheuristic for the SZP in this paper, our focus of the experiment will concentrate to give an understanding on the behavior and performance of the quality measurement of the proposed multiobjective metaheuristic algorithm. We consider 3 zones created for each zoning plan and we use 55 basic units for the input of the model. Then, we conduct an experiment to compare the result of the objectives achieved with three different quality measurement methods for the acceptance rules as below:

- Minimization of Achievement Tchebycheff Scalarizing Function (ATSF)
- Minimization of Weighted Tchebycheff Scalarizing Function (WSF) with $S(z,z^0,\Lambda)=\max_j\{\lambda_j(z_j^{\,0}-z_j)\}$
- Minimization of a weighted additive multicriteria function (WAMCF) $F(x)=\sum_r \alpha_r f_r(x)$, where α_r is a weight and $f_r(x)$ is the value of a function assigning a value of criterion r to any given solution x.

The use of the first two measurement methods is to compare the influence of different acceptance rules on overall performance of the multiobjective metaheuristic method. Meanwhile, the use of the third minimization function is to proof the effectiveness of the proposed multiobjective metaheuristic method in multiobjective environment rather than in a common multicriteria environment with a weighted additive multicriteria function.

Table 1 shows a comparison of the objectives functions for the original zoning plan with the improved plans generated from the above three minimization methods. From the result, we can easily observe that the solutions from the proposed algorithm helps optimize all objectives simultaneously and the feasible solutions yielding the best compromise among all objectives on a set of Pareto-optimal solutions both in case one and two. However, for case three, the outputs are worse than the original zoning plans especially in the first objectives.



Objective	Weight vector
f_1	0.6
f_2	0.3
f_3	0.1

Fig. 3. Initial zoning plans with basic units and weight vectors assigned for each objective

Table 1. Objective functions values for solution compare to the initial solution

Objectives	Original plan	Improved plans with ATSF		Improved plans with WSF		Improved plans with WAMCF	
		1	2	1	2	1	2
f_1	1.3875	0.7285	1.3047	1.1312	1.3047	***1.5651***	***1.7332***
f_2	0.9840	0.9837	0.9809	0.9839	0.9809	0.9823	0.9827
f_3	8.3516	7.6705	5.2255	7.5682	5.2255	4.9587	5.4424

4 Conclusion

State of the art of multiobjective optimization for spatial zoning problem is a practical solution. This study has formulated this problem as a multiobjective one and it has developed a hybrid metaheuristic scheme to solve a realistic multiobjective spatial zoning problem. GIS tool is employed in data acquisition, storage, retrieval, and management of the spatial data for the use of the multiobjective metaheuristic method in making spatial zoning decisions. The hybrid tabu scatter search helps in improving the multiobjective optimization part in term of better approximation of the non-dominated set and wide exploration of the solution search with intensification and diversification techniques. Preliminary numerical experiments validate our method with instance of three most commonly defined objectives in generating a simple three-zones plan. The results are promising for our method from the experiment of different quality measurement methods used.

References

1. Wang Y. C., Bong C. W.: Compactness Measurement Using Fuzzy Multicriteria Decision Making for Redistricting. In Proceeding IEEE REGION 10 International Conference on Electrical and Electronic Technology. (2001) IEEE Press.
2. Bozkaya B., Erkut E. and Laporte G.: A Tabu Search Heuristic and Adaptive Memory Procedure for Political Districting. European Journal of Operational Research 144, (2003).
3. Agnew J.: Geography and Regional Science Program: Geographic Approaches to Democratization. National Science Foundation. Washington. (1994)
4. Altman M.: Redistricting Principles and Democratic Representation. Ph.D. Thesis; California Institute of Technology; Pasadena, California. (1998)
5. Landa-Silva J. D. and Burke E.K.: Multiobjective Metaheuristics for Scheduling and Timetabling. University of Nottingham, UK [Online Tutorial]. (2002) URL: http://tew.ruca.ua.ac.be/eume/welcome.htm?workshops/momh/
6. Knowles J. D.: Local-Search and Hybrid Evolutional Algorithms for Pareto Optimization. PhD Thesis, Department of Computer Science, University of Reading, Reading, UK. (2002)
7. Openshaw S.: Developing GIS-relevant Zone-based Spatial Analysis Methods", Chapter 4 of Longley P. and Batty M.(eds), Spatial Analysis: Modeling in a GIS environment. Geo-Information Intenational, Cambridge. (1996)
8. Ehrgott M.and Gandibleux X. A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimisation. OR Spektrum 22, 425 –460, Springer-Verlag. (2000)
9. Church R. L., Sorensen P.: Integrating Normative Location Models into GIS: Problems and Prospects with the p-median Model. Technical Report 94-5, NCGIA. National Center for Geographic Information and Analysis. (1994)
10. Jain A. S.: A Multi-Level Hybrid Framework for the Deterministic Job-Shop Scheduling Problem. PhD Thesis, Department of Applied Physics, Electronic & Mechanical Engineering University Of Dundee, Dundee, Scotland, U.K. (1998)
11. Hansen M.P.: Tabu Search for Multiobjective Optimization: MOTS. Proceedings of MCDM'97. (1997)
12. Yagiura M. and Ibaraki T.: On Metaheuristic Algorithms for Combinatorial Optimization Problems. The Transactions of the Institute of Electronics, Information and Communication Engineers, Vol. J83-D-I, No.1, 3-25. (2000)

Dynamic User Profiles Based on Boolean Formulas

Czesław Danilowicz and Agnieszka Indyka-Piasecka

Wroclaw University of Technology, Department of Information Systems,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{danilowicz, indyka}@pwr.wroc.pl

Abstract. In this paper the model of interests and preferences of information retrieval system users is presented. The paper starts with a short overview on areas of user profiles applications. Subsequently, a method to represent user profiles in the field of Web document retrieval by using query terms and weighted terms of retrieved documents is presented. The method is based on evaluating the relevance of retrieved documents by the user. The relevant documents are the basis to construct so called subprofiles. The created subprofiles represent user interests and preferences, and are used for the user query modification.

Keywords. Internet Applications, Intelligent Systems

1 Introduction

Small, constant in time and homogeneous collections of documents are characteristic for the classical information retrieval. But these features are not valid for information resources stored in the Word Wide Web. The number of Web sites is huge and is still growing quickly. The number of WWW pages in the February 1999 was estimated at about 800 millions [5], and in the August 2003 there were 3.000 millions Web pages indexed by the Goggle engine. In the year 2000, over 42 millions of servers were working at the Web and the network was being used by over 200 millions of users (Netcraft Services [20]).

It is search engines, which are the basic tool used by 85% of Web users for retrieving important information from the WWW [18]. But using the engine is not so obvious for an average user. The lack of evident order of relevant documents is one of the most important disadvantages of Web search engines noticed by the users.

In the classical documents collection, the document description is usually normalized. There are introduced special fields of records for the purposes of retrieval, e.g. descriptors, keywords, or classification keywords. These fields should be filled by authors obligatorily. However, it is not the case for the Web resources. In [18], it is claimed that the author's keywords are only used on the pages of 34% of sites.

The Web users always want to maximally satisfy their own information needs, putting a minimal effort to formulate more detailed queries. It was not possible in case of

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 779-787, 2004.

the classical information retrieval systems. However, in the Web, short and broad-topic queries, introduced by the users [22] together with not very precise indexes have caused that retrieval quality has been decreasing additionally [16].

In the search engine, regardless of the pages indexing method or the data structure used, the answer for the user query is a list of documents, ordered accordingly to the estimated relevance weight (weight RSV). The quality of an answer is usually not satisfied enough [4][11][17], but users can evaluate an ordered list of documents better than unordered answer [2][7][13][14] because the more relevant documents are placed rather at the beginning of the list, than at the end of it. This "partial order" delivers less information, which is needed to estimate the relevance of documents, than the parallel ranking, provided in the classical information retrieval systems, did. In these systems, the description of documents presented to the user is detailed enough to estimate the relevance of documents. Conversely, information about Web pages presented in search engines is poor and various, what forces the user to open the particular pages of the answer to estimate their relevance. Many of these pages appears to be not relevant according to the user information needs [1][6][8].

The problems of the information retrieval in the Web, described above, have stimulated researchers to search for new approaches. One of them is *modeling of user information needs*. The model of user information needs is usually called *profile*. The user profile, firstly, has been introduced in systems of *Selective Distribution of Information* (SDI), as a new element in information retrieval systems. In SDI systems, the user profile contains the representation of user interests and preferences, which have been stayed unchanged for a period of time. The user profile was being determined during process of information retrieving. In SDI systems defining the user profile and introducing data into it have been easier than in Web search engines. Also the ways of changing the collection of documents were determined, and the user query stayed unchanged being permanently applied for filtering of documents [6][19]. Contrary, in the reality of the Web, the subsequent queries asked by the user are varied and Web resources (collection of documents) are changing quickly. Thus, the profile of Web user should evolve dynamically, too.

In both cases, the user profile is a secondary, but important, source of information about user information need expressed through the query. In that way, using the user profile corresponds to the idea introduced by Fox [10]:

> *Effective integration of more information should lead to better information retrieval.*

However, some authors underline, that representation of user information need dos not include other aspects of retrieval process. They propose to take into consideration, as parts of the representation of user profile, the other aspects, such as: the number of interface elements user can manipulate simultaneously [3][24], ability to see different colors [12], or the level of knowledge acquired in domain of user interests [9] But these are only postulates and the research in these domains is at preliminary phase, yet. The main problem are technical and practical limitations that affect experiments [23]. However, these problems do not prevent from continuing research on modeling of IRS users, because the representation of user information needs must be privileged and treated as the primary goal in the area of information retrieval. The other aspects, mentioned above, concern mostly interface, and require different research methods.

2 Elements of the Model of Information Retrieval System

Information retrieval system is defined by the following elements:

1. *The following sets:*

a) T – set of terms (the dictionary)
b) D - set of documents,
c) Q - set of queries,
d) U - set of users,
e) P - set of profiles.

2. *Retrieval function:*

$$\omega:Q\rightarrow 2^D,$$

where its value, i.e. $D_q=\omega(q)$, is the answer of the retrieval system for the query q (i.e. the set of retrieved documents).

3. *Relevance function*:

$$r:(D\times U)\rightarrow 2^D,$$

where $D_q^u=r(D_q,u)$ is the set of relevant documents pointed out by the user u from the answer for the query q. Therefore, we have $D_q^u \subseteq D_q$.

Profiles from the set P are determined by the profile modification function π as following:

$$p_{i+1}^u = \pi(p_i^u)$$

where:
p_i – profile of the user u before modification,
p_{i+1} – profile of the user u after modification.

In the next section, we will introduce profile modification as a composition of subprofiles modification. During the process of modification, the subprofile is deleted from the profile if it has not been modified during time t preceding the last query asked by the user.

3 Structure of Documents

The dictionary is the set of N terms:

$$T = \{t_1, t_2, ...,t_N\}$$

A document d is represented by a n–dimensional vector:

$$d = (d_1, d_2, ...,d_N),$$

where d_i is the weight of term t_i in document d. The weight of a term is equal 0 if this term does not appear in the document.

A query is a Boolean expression:

$$q = q_1 \wedge q_2 \wedge \ldots \wedge q_T,$$

where $q_i = t_i$ or $q_i = 1$ (1 is a logical value).

For instance, if $T = \{t_1, t_2, t_3, t_4\}$, the valid queries are:

$q_a = 1 \wedge t_2 \wedge 1 \wedge 1 = t_2$,

$q_b = t_1 \wedge 1 \wedge 1 \wedge t_4 = t_1 \wedge t_4$,

User profile, for the user u_i, is a set of subprofiles:

$$p^{u_i} = \{s_1^{u_i}, s_2^{u_i}, \ldots, s_{M_i}^{u_i}\}$$

To make descriptions less complicated, we will abandon some symbols and indexes, where it should not cause any ambiguity. Following this rule, user profile will be described as a set:

$$p = \{s_1, s_2, \ldots, s_M\}$$

Subprofile is a triple:

$$s = <i,v,l>,$$

where:

i – subprofile identifier, a Boolean expression that identifies subprofile,

v – vector of weights of terms that represents subprofile,

l – the identifier of the last modification.

Example 1:

$T = \{t_1, t_2, t_3, t_4\}$.

The following expression is compatible with the requirements of the user profile definition:

$p = \{s_1, s_2, s_3\}$,

where

$s_1 = < t_1 \wedge t_4$, (0.6, 0.1, 0.4, 0.7), 15>,

$s_2 = < t_1 \wedge t_4$, (0.5, 0.1, 0.1, 0.6), 3>,

$s_3 = < t_2 \wedge t_3$, (0, 1, 1, 0), 0>.

We will use the notions of *broader query* and *narrower query* (a Boolean expression) while describing process of query creation and modification and profile modification. The interpretation of these notions is as following:

Let $q_a = q_a^1 \wedge q_a^2 \wedge \ldots \wedge q_a^K$, and $q_b = q_b^1 \wedge q_b^2 \wedge \ldots \wedge q_b^M$. We say, that *query q_a is broader than query q_b*, when $\{q_a^1, q_a^2, \ldots, q_a^K\} \supset \{q_b^1, q_b^2, \ldots, q_b^M\}$, and we say, that *query q_a is narrower than query q_b* when $\{q_a^1, q_a^2, \ldots, q_a^K\} \subset \{q_b^1, q_b^2, \ldots, q_b^M\}$. The relations defined above will be written as $q_a \geq q_b$ and ($q_a \leq q_b$), respectively.

4 Query Modification

Interests of user of the Web information system usually do not focus on one topic but concern the various problems. Therefore the user introduces queries that represent various subjects (topics) - or even various domain of knowledge - into the system. The well known classical, homogeneous IRS profile, where all queries and results of queries concerning various domains are registered together, is inappropriate to be used for the Web systems. The modifications of various queries made with this profile, that represents different interests, lead to unpredictable influence of interests in one domain on the retrieval results in the other domain. Such problems will not occur for the user profile structure, in which different interests are represented by different subprofiles [15]. Subprofile is a register of the information about user interests and preferences, and is univocally associated with the user query.

The particular elements of the subprofile get the following interpretation:

The vector of weights of terms v represents user information needs in subprofile s. The higher the weight of term is, the more interested in subject represented by this term the user is. And the small weight means that user is not interested in subject represented by this term. The threshold τ is introduced to distinguish between terms of (relatively) *high* and *low* weights. The subprofile identifier i is a conjunction of terms, weights of which in vector v are over the threshold τ. Query q will be modified by subprofile s when $q\geq i$. Query modification is understood here as "to replace" query q by subprofile identifier i. The modified query is the same as identifier i, and the set of documents $D_i = \alpha(i)$ is the answer of information retrieval system for modified query.

If, for the given query q asked by the user, the condition $q \leq s$ is valid for more than one subprofile, the answer for that query is the sum of the answers for all modified queries, which have been created on the base of all matching subprofiles.

If none of subprofiles matches the modification condition for the given query q, a new subprofile is introduced into the profile.

Example 2:

$T = \{t_1, t_2, t_3, t_4\}$,

$\tau = 0.4$

$p = \{s_1, s_2, s_3, s_4\}$,

$s_1 = \langle i_1, v_1, l_1 \rangle = \langle t_1 \wedge t_4,\ (0.6, 0, 0.1, 0.7),\ 3 \rangle$,

$s_2 = \langle i_2, v_2, l_2 \rangle = \langle t_1 \wedge t_3 \wedge t_4,\ (0.5, 0.1, 0.5, 0.9),\ 8 \rangle$

$s_3 = \langle i_3, v_3, l_3 \rangle = \langle t_2 \wedge t_4,\ (0, 0.5, 0.1, 0.8),\ 2 \rangle$,

$s_4 = \langle i_4, v_4, l_4 \rangle = \langle t_4,\ (0.2, 0.1, 0.1, 0.9),\ 11 \rangle$.

$q_a = t_1 \wedge t_4$,

$q_b = t_1 \wedge t_2 \wedge t_4$,

$q_c = t_2 \wedge t_3$.

The query q_a will be modified by the subprofiles s_1 and s_4, and the query q_b –only by the subprofile s_3. The query q_c will not be modified, but a new subprofile will be added.

5 Profile Modification

Modification of the profile p represents changes of user interests and preferences. The information concerning these changes is included, in some way, in the query q. The query can modify one or more subprofiles, it is associated to, or, when a query represents a new subject, a new subprofile is added into the profile. The subprofile *modification* or *addition* are the basic operation of the profile modification process.

1. Subprofile $s = <i,v,l>$ is modified when:

   - the query q is narrower than the identifier i,

   - and the last subprofile modification has been made during processing of one of the m last user queries.

   The identifier i replaces user query q and retrieval with this new query is performed, what starts modification process. So, the result of retrieval is the set $D_i = \sigma(i)$. User points out these documents which he considers relevant for him in the set of retrieved documents. A subset $D_i^u = r(D_i, u)$ is marked. Then, the representation of the set D_i^u is calculated. This representation is vector $w = (w_1, w_2, ..., w_N)$, where w_i represent the importance of the term t_i^z in the set D_i^u. In the modified subprofile $s' = <i',v',l'>$, element v' represents the merge of vectors: v and w. The identifier i' is the conjunction of these terms the weights of which are over threshold τ in the vector v'. The number $l'= 1$, because it represents the last modification made.

2. If the query q is not narrower than any identifier i in profile p, the new subprofile is added into the profile. The elements of the new subprofile are as follows: $i = q$, weights in the vector v are set to 1 for terms that appear in user query and to 0 for other terms, and $l = 1$.

3. If subprofile $s = <i,v,l>$ was not modified by the query q, only the identifier l of the last modification is increased by l, i.e. $s' = <i,v,l+1>$.

If during m queries asked by the user, no modification of the subprofile $s = <i,v,l>$ was made, the subprofile is deleted from the profile p.

5 Vectors Merging

Merging of vectors happens in the following three cases:

1. When the representation of the set of relevant documents is calculated,
2. When the retrieval results of modified queries are being joined,
3. When the vector representing the set of relevant documents is being merged with the term weight vector representing subprofile.

Merging of vectors proceeds as following:

Let $a_1 = (a_1^1, a_2^1, ..., a_N^1)$, $a_2 = (a_1^2, a_2^2, ..., a_N^2)$,…, and $a_S = (a_1^S, a_2^S, ..., a_N^S)$ be vectors. The result of merging of the vectors is the vector $A = (a_1, a_2, ..., a_N)$, where

$$a_j = \frac{a_j^1 + a_j^2 + ... + a_j^S}{S} \quad \text{for } j = 1,2,\ldots,N.$$

Example 3:

$T = \{t_1, t_2, t_3\}$,
$D = \{d_1, d_2, d_3, d_4\}$,
$d_1 = (0.4, 0.1, 0.7)$,
$d_2 = (0.1, 0.2, 0.9)$,
$d_3 = (0.6, 0.7, 0.1)$,
$d_4 = (0.5, 0.2, 0.9)$,

$p = \{s_1, s_2, s_3\}$,

$s_1 = <t_1 \wedge t_2,\ (0.1, 0.4, 0.1),\ 20>$,

$s_2 = <t_3,\ (0.1, 0.2, 0.7),\ 3>$

$s_3 = <t_2 \wedge t_3,\ (0.1, 0.3, 0.4),\ 5>$,

Threshold $\tau = 0.3$ and $m = 20$.

Let the following query is being asked by the user:

$q = t_1 \wedge t_3$,

The query is replaced by the identifier of subprofile s_2, i.e. by $i = t_3$.

$D_i = \alpha(i) = \{d_1, d_2, d_4\}$.

Relevant documents pointed out by the user are: d_2 and d_4.

The vector (0.6, 0.4, 0.7) is the representation of relevant documents, pointed out by the user.

The vector (0.2, 0.2, 0.8) is the result of merging two vectors, namely: (0.3, 0.2, 0.9) and (0.1, 0.2, 0.7).

After modification of the subprofile, s_2 will have the following form:

$s_2' = <t_3,\ (0.2, 0.2, 0.8),\ 1>$.

Subprofile s_1 has been deleted, and the counter l is increased in the subprofile s_3.

Finally, the state of the profile after modifications will be as following:

$p = \{s_2', s_3\}$,

$s_2' = <t_3,\ (0.1, 0.2, 0.7),\ 3>$

$s_3 = <t_2 \wedge t_3,\ (0.1, 0.3, 0.4),\ 6>$.

6 Conclusions

In the information retrieval method presented in this paper, the profile plays the role of a kind of „background" for the queries asked by the user. The only possibility to modify a query is to broaden it because queries play a dominant role in retrieval process and a broader query causes additional documents to be found. Subjects of these documents are connected with the queries previously asked by the user to the system. The importance of this background (i.e. subprofiles) could be regulated in two ways. Firstly, we can, by means of parameter *m*, enlarge or shorten the period of time whereby out-of-use subprofile is removed. Secondly, we can also increase or decrease the influence of the questions on subprofiles. In order to achieve this, we need to introduce a parameter into the merging operation. This parameter would control the influence of retrieved documents on the modification of a subprofile. The relationship between query, profile and retrieved documents could also be regulated by means of the consensus methods [21]. More in–depth research and experimental verification in limited collections of documents as well as in real WWW environment are necessary to be done.

References

1. Belkin N.J., Croft W.B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin, Communications of the ACM, 35(12), (1992), 29–38.
2. Belkin, N. J., Croft, W. B.: Retrieval Techniques, Annual Review of Information Science and Technology, 22(9), (1987), 109-145.
3. Brusilowski, P., Schwarz, E.: User as Student: Towards an Adaptive Interface for Advanced Web–Based Applications, Proc. of the 6th International Conference on User Modeling, UM'97, Sardinia, Springer Vien New York, (1997), 177–188.
4. Butler, D: Souped-up search engines, Nature, 405, (6783), . (2000), 112-115.
5. Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J. M., Gibson, D.: Hypersearching the web, Scientific American, June (1999).
6. Daniłowicz, C.,: Model of information retrieval systems with special regard to user' preferences. Monographs No.3. Wroclaw University of Technology Press (1992).
7. Daniłowicz, C., Nguyen, H.C., Nguyen, N.T.: Model of Intelligent Information Retrieval Systems using User Profiles. Proceedings of the Conference on Business Information Systems BIS 2003, Colorado Springs, USA (2003).
8. Daniłowicz, C., Nguyen, H.C.: Using User Profiles in Intelligent Information Retrieval. Proceedings of ISMIS 2002 Conference, LNAI 2366, Springer-Verlag, (2002), 223-231.
9. De Carolis, B., Pizzutilo, S.: From Discourse Plans to User–Adapted Hypermedia, Proc. of the 6th International Conference on User Modeling, UM'97, Sardinia, Springer, Vien New York, (1997), 37–40.
10. Fox, E. A., Nunn, G. L., Lee, W. C.: Coefficients for combining concept classes in a collection, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, (1988), 291-307
11. Gordon, M., Pathak, P.: Finding information on the World Wide Web: the retrieval effectiveness of search engines, Information Processing and Management, 35(2), (1999), 141-180.
12. Gutkauf, B., Thies, S., Edwards, A. D. N.: A User Adaptive Chart Editing System Based on User Modeling and Critiquing, Proc. of the 6th International Conference on User Modeling, UM'97, Sardinia, Springer Vien New York, (1997), 159–170.
13. Harman, D.: Relevance Feedback Revisited, SIGIR'92, Proceedings of the Fifteenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Denmark, ACM Press, (1992), 1–10.
14. Harman, D.: Ranking algorithms, In: Frakes, Baeza-Yates (ed.) Information Retrieval - Data Structures & Algorithms, Prentice Hall, Englewood Cliffs, New Jersey, Chapter 14, (1992), 363-392.
15. Indyka-Piasecka, A., Piasecki, M.: Adaptive Translation between User's Vocabulary and Internet Queries, Proceedings of the International Conference on Intelligent Information Systems, New Trends in Intelligent Information Processing and Web Mining: IIPWM'03, Zakopane, Poland, Springer, (2003), 149–157.
16. Jansen, B., Spink, A., Bateman, J., Saracevic, T.: Real life information retrieval: a study of user queries on the web, SIGIR Forum, 32(1), (1998), 5-17.
17. Kobayashi, M., Takeda, K.: Information Retrieval on the Web, ACM Computing Surveys, 32(2), (2000), 144-173.
18. Lawrence, S., Giles, C.L.: Accessibility of Information on the Web, Nature, vol. 400, (1999), 107–109.
19. Myaeng, S.H., Korfhage, R.R.,:Integration of User Profiles: Models and Experiments in Information Retrieval. Information Processing & Management, 26, (1990), 719-738.

20. Netcraft., http://news.netcraft.com/archives/web_server_survey.html
21. Nguyen, N.T., Sobecki, J.: Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems. Universal Access in Information Society 2(4) (2003) 342-358 (Springer-Verlag).
22. Silverstein, C., Henzinger, M., Marais, J., Moricz, M.: Analysis of a very large AltaVista query log, Technical Report TR 1998-014, Compaq Systems Research Center, Palo Alto, CA (1998).
23. Sobecki, J.: XML-based Interface Model for Socially Adaptive Web-Based Systems User Interfaces. Lecture Notes in Computer Science 2660 (2003) 592-598.
24. Strachan, L., Anderson, J., Sneesby, M., Evans M.: Pragmatic User Modeling in A Commercial Software Systems, Proc. of the 6th International Conference on User Modeling, UM'97, Sardinia, (1997) 189–200, Springer Vien New York.

Application of Intelligent Information Retrieval Techniques to a Television Similar Program Guide

Chandrashekar Machiraju, Sirisha Kanda, and Venu Dasigi

School of Computing and Software Engineering
Southern Polytechnic State University
1100 South Marietta Pkwy
Marietta, Georgia 30060, U.S.A.
{cmachira, skanda, vdasigi}@spsu.edu

Abstract. The advent of Digital Television and the “Connected Home” has led to an information overload to the viewer. A lot of research has been conducted towards developing interactive program guide services and products that attempt to simplify the television viewing experience. Closed captions that are part of the transmission signal of the programs aired have been identified as a good source of information [1]. Information Retrieval techniques may be applied to closed caption transcripts to create applications such as the one we describe here, called the “Similar Program Guide (SPG)”. Such a guide would notify the user of programs being aired on other channels, similar to the one he/she is currently watching or similar to the ones in the user’s profile. Additionally, this guide would allow users to compile their favorite programs and thus maintain user profiles. In this paper we discuss the approaches for development of the application. We use Information Retrieval techniques as the basis for this application.

1 Introduction and Background

The introduction of digital television has led to an ever-increasing number of channels available to the consumer. The number of channels coupled with the wide range of programs aired, makes it extremely difficult for the average viewer to keep track of the schedules of programs he/she likes. A partial solution to this problem has been provided by means of □Electronic Program Guides (EPG□s)□ An EPG is an electronic version of the printed television guide which displays the schedules of programs to be shortly aired. However, there is a need for additional tools that allow the viewer to cope with the mass of content available through the television.

A □Similar Program Guide (SPG)□would notify the user of any programs being currently aired that are similar to the one□s he/she is watching. If the user selects any of the suggested programs, it is added to the user profile. Alternatively, the SPG notifies the user of any programs, similar to the ones in his/her profile that are currently being aired. In this context, a user profile is a collection of closed caption transcripts of programs liked by the viewer. We use content-based recommendation strategy where items similar to those a user has liked in the past are recommended.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 788-796, 2004.

Our approach is to be contrasted to collaborative recommendation or social filtering which is another strategy, where items are recommended because they were liked by similar users [2]. However, the performance of the system depends on the number of users using it. An analysis of the users rather than the items is carried out by the recommender [3].

Closed captions (CC) are captions that are hidden in the video signal in television and cable transmissions, invisible without a special decoder. The closed captions are hidden in line 21 of the vertical blanking interval. XDS (eXtended Data Services) data are also sometimes stored in a field of line 21.

Most television sets have a built-in closed caption decoder, and almost all television programs come with closed captions, nowadays. These captions have a number of applications [1]. These captions can be used for content-based management of user-profiles and for implementing the functionality of the SPG. We use text retrieval and text classification techniques as the basis for the development of the SPG.

Some issues pertaining to the retrieval and classification of closed caption transcripts have been identified, most significant of which is the dialog nature of the transcripts as opposed to the narrative nature of the text that has been traditionally used in the field of Information Retrieval. More specifically, closed caption transcripts, especially of movies, sports programs and commercials, contain dialogs while the traditional text documents such as the ones found on the World Wide Web, for example, describe a specific topic of interest [1]. These issues tend to degrade retrieval and classification performance. This distinction of the dataset signifies the need for evaluating the performance measures of the existing algorithms.

The existence of a wide range of retrieval and classification algorithms calls for identifying the best algorithm for our purpose. We have used □off-the-shelf□algorithms available through the bow library [4] for comparing the performance measures of the existing retrieval and classification algorithms. We have found that our initial experiments for designing the Similar Program Guide have been encouraging. In this paper, we present these results as evidence of the promise of this approach. The rest of the paper is organized as follows. Section 2 describes two approaches we have identified for the implementation of SPG; in Section 3 we describe the experimental setup; Section 4 describes the procedure used for conducting the experiments, followed by the presentation of results and their analysis in sections 5 and 6 respectively. We conclude with an indication of the direction of ongoing research.

2 Approaches

The SPG can be implemented using either a classification-based approach or a retrieval-based approach. These approaches are somewhat different in the results that they produce. We present a brief description of these approaches in these sections.

2.1 Classification-Based Approach

The classification-based approach requires the programs being aired to be classified to pre-defined categories. The captions of a program that a user is watching and of programs on other channels are captured simultaneously. On a query by the user for currently aired programs similar to the one he/she is watching, these captions that are being captured are classified to the pre-defined categories. Information (program name, channel name, etc.) of programs that are classified to the same category as the program the viewer is currently watching is classified, is displayed. The system suggests the user to add the current program to his/her profile if it already does not exist in the profile. Thus the system also builds a user profile that can be used to recommend similar programs in future.

The success of this approach depends on the ability to classify the captions precisely. A number of classification algorithms exist that have shown varied performance measures. Among others, naïve Bayes and Support Vector Machines are two of the common methods for effective classification. However, the dialog nature of the problem set and the presence of transmission and transcription errors call for a careful evaluation of performance measures of the classification methods with respect to closed captions. The results of these experiments are shown in Section 5.

2.2 Retrieval-Based Approach

As stated earlier, the performance of the classification-based approach is implicitly tied to the manner in which captions are pre-classified. Categories can be very broad and hence may include programs that have subtle differences. In other words, the classification-based approach could suffer from a possible lack of similarity in the content of the query document and the documents belonging to its category. For example, a court room drama (example "The Practice") and an investigative show (example "CSI Miami") may both be classified into a possible "Crime" category, but differ significantly in their content. On the other hand, two programs that share certain similarities may be classified into different categories, if the categories are too narrow (although not in our case).

The retrieval-based approach addresses these issues by eliminating the need for classifying the captions and bases the results on the content of the query document and the documents in the corpus. More precisely, in this approach the captions of the program a user is currently watching are used as the query and captions of other programs currently aired form the corpus. A ranked list of documents similar to the query is generated and program information (program name, channel name, etc.) of the top several documents would be displayed. The idea is similar to the traditional "query by example" or "similar pages" link in some search engines. We use the Vector Space Model for this approach and the results are shown in the Section 5.

3 Experimental Setup

We use standard text classification and retrieval techniques to study these issues. We classify closed caption transcripts in two ways: First we employ high-level or coarse-grained categories, such as *news*, *movies*, *sports*, and *commercials*. Then within the movies category, we further classify them into genres such as, *action*, *adventure*, *comedy* and *drama*. We consider these as low-level or fine-grained categories.

With our focus mainly on evaluating the performance of na□e Bayes and Support Vector Machines when applied to closed caption transcripts, we have initially decided to make use of some standard classification and retrieval tools, rather than create new algorithms. We use the *bow* toolkit with the *Rainbow* front-end for classification and the arrow front-end for retrieval. This toolkit was developed by Andrew McCallum, et al from Carnegie Mellon University [4]. *Bow* is a library of C code useful for writing statistical text analysis, language modeling, and information retrieval programs. *Rainbow* is a front-end which uses *Bow* library to perform statistical text classification. Several different *classification* methods are available in *Rainbow*, including na□e Bayes and Support Vector Machines, which we use in our experimentation. *Arrow* uses the bow library for TF-IDF-based *retrieval*. The general pattern of using *Rainbow* or *Arrow* involves two steps: i) First, *Rainbow/Arrow* reads the training documents and writes to disk a □model□ containing their statistics. ii) Next, using the model, *Rainbow/Arrow* performs classification/retrieval on test documents.

3.1 Capturing and Preprocessing Closed Caption Transcripts

There are several devices that can capture the text of the captions into a computer file. They may be plug-in cards or external devices. We captured our news transcripts, movies, sports, and commercials from various transmissions. Because the captions end up as a long file they have to be preprocessed before continuing.

In order to do the classification experiments, the captured closed caption text was separated into individual files corresponding to the □program units□ first. After observing the captured data thoroughly, we found a pattern between consecutive television program transcripts that could be used to separate them. At the end of each program, credits are given to the closed captioning service. There almost always are the strings □caption□, □captioned□, □captions□, □captioning□, or their variants after each program. We developed a program to do the separation, which worked well for our purposes.

There were no standard data sets of closed caption transcripts available to the authors□knowledge. Therefore, we had to both generate the data sets and manually classify them. For the high-level/coarse-grained classification - classification of the transcripts as commercials, movies, sports and news - the individual files were manually classified by a cursory reading of the documents. This manual classification generated the corpus for the high-level/coarse-grained experiments which contained 50 transcripts in the commercials category, 100 transcripts in the movies category, 50 in the sports and 50 in the news category.

For the low-level/fine-grained classification - classification of movie transcripts to action, adventure, comedy and drama - we first read the movie transcripts to get their titles. Then we searched an existing movie classification database[1] on the web to get their subcategories. This manual classification of the movies category from the high-

[1] http://www.filmratings.com/

level categories generated the corpus for our low-level/fine-grained classification which contained 25 transcripts in each of the four low-level categories.

4 Experimental Procedure

4.1 Classification of Closed Caption Transcripts

Cross validation is used in our experimentation to avoid sensitivity to specific choices of training data [5]. Results are averaged over 50 trials, each time using 60% transcripts randomly chosen, as the training set and the other 40% as the test set. The experiment was repeated for both levels of classification using na□e Bayes and SVM algorithms.

4.2 Retrieval of Closed Caption Transcripts

Individual closed caption transcripts formed the corpus for the purpose of experimentation. One of the documents was used as the query and the top-10 precision of the retrieved documents was analyzed. A retrieved document was considered relevant if its content matched the content of the query. For instance, on a query about car races, retrieved documents about dog races, marathons etc. were considered to be relevant. We admit that judging similarity in this manner is subjective. However, we expect this level of subjectivity among real users. We use the same corpus that was used for classification in our retrieval experiments as well and occasionally refer to the categories of documents/transcripts in the context of retrieval too, although these categories are not directly relevant for the purposes of retrieval. This methodology allows us to compare the results generated by the classification-based and retrieval-based approaches. For instance, in the previous example, it is conceivable that a transcript related to a car race could be in the *sports* category or in the *movies* category.

5 Results

5.1 Classification Results

Precision and recall on each category and macro- and micro-averaged versions of them were calculated [6]. To review the definitions briefly, if tp_i, fp_i, tn_i, and fn_i stand respectively for the numbers of true positives, false positives, true negatives, and false negatives for the i-th category, we have:

$$\text{precision}_i = tp_i / (tp_i+fp_i)$$
$$\text{recall}_i = tp_i / (tp_i+fn_i)$$

There are two averaging methods. *Macro-averaging* refers to the usual method of averaging the individual measures across all categories. In *micro-averaging*, the grand total aggregates of the numbers for true/false positives and false negatives are computed and the appropriate ratios give us the metrics as follows, *with* i *ranging over all categories*:

$$\begin{aligned}
\text{Macro-averaged precision} &= \Sigma_i \text{ precision}_i / N \\
\text{Macro-averaged recall} &= \Sigma_i \text{ recall}_i / N \\
\text{Micro-averaged precision} &= \Sigma_i \text{ tp}_i / \Sigma_i (\text{tp}_i+\text{fp}_i) \\
\text{Micro-averaged recall} &= \Sigma_i \text{ tp}_i / \Sigma_i (\text{tp}_i+\text{fn}_i)
\end{aligned}$$

Tables 1-4 show our classification results. Tables 1 and 2 respectively show summaries of our results on coarse-grained and fine-grained classification using the na□ve Bayes approach. Tables 3 and 4 similarly show the results of SVM-based classification. As indicated in Section 4.1, each of these tables represents results averaged over 50 trials, for good cross validation.

Table 1. Coarse-grained/High-level classification results using na□ve Bayes

	Average %	Standard Deviation %
Macro precision	95.387	3.177
Macro recall	83.356	4.892
Micro precision	91.065	2.925
Micro recall	91.065	2.925

Table 2. Fine-grained/Low-level classification results using na□ve Bayes

	Average %	Standard Deviation %
Macro precision	56.44	8.93
Macro recall	59.82	9.59
Micro precision	51.60	8.90
Micro recall	58.79	9.46

5.2 Retrieval Results

For retrieval, we calculated the top-10 precision for each query. The results of retrieval are often sorted in decreasing order of a score that reflects the similarity between the query and the retrieved item. A user is generally not interested in (nor has the time to look at) more than the first ten or twenty items from such a long list. The *top-10 precision* measures the fraction of the items ranked among the top ten retrieved items that are actually relevant to the query. Tables 5 and 6 show the results of retrieving similar programs based on content, from the high-level data and low-level data, respectively.

Table 3. Coarse-grained/High-level classification results using SVM

	Average %	Standard Deviation %
Macro precision	97.13	2.11
Macro recall	93.67	4.21
Micro precision	96.52	2.14
Micro recall	95.52	2.14

Table 4. Fine-grained/Low-level classification results using SVM

	Average %	Standard Deviation %
Macro precision	74.40	16.31
Macro recall	69.60	11.43
Micro precision	69.60	13.76
Micro recall	74.56	12.11

Table 5. Retrieval results from High-level data

Query	Test 1	Test 2	Test 3	Average %
	Top -10 Precision %	Top -10 Precision %	Top -10 Precision %	
Commercial	100.00	100.00	100.00	100.00
HBO	100.00	100.00	100.00	100.00
News	100.00	100.00	100.00	100.00
Sports	100.00	100.00	100.00	100.00

Table 6. Retrieval results from Low-level data

Query	Test 1	Test 2	Test 3	Average
	Top -10 Precision	Top -10 Precision	Top -10 Precision	
Action	60.00%	70.00%	60.00%	63.33%
Adventure	50.00%	60.00%	60.00%	56.67%
Comedy	70.00%	70.00%	50.00%	63.33%
Drama	50.00%	50.00%	50.00%	50.00%

6 Discussion of Results and Conclusions

On comparing the results in Table 1 (for the na□ve Bayes approach) to those in Table 3 (for SVMs), for high-level classification, we observe no significant difference in performance. Here, SVMs show a slightly better performance with less deviation from the average. However, on comparing the results from the low-level classification in Table 2 (for na□ve Bayes) to those in Table 4 (for SVMs), we observe that SVMs perform clearly better on the average.

Table 5 presents the retrieval results at the high level, and Table 6 presents the retrieval results at the low level. In the retrieval-based approach a document with □similar content□ was considered relevant even though it belonged to a different category. For each query that we tested at the high level, all the documents retrieved had similar content and hence, the results here are 100% for all the tests. We observe, and believe it is worthy of note, that there were some documents retrieved as similar based on content, although they belonged to categories different from that of the query. These cases would be missed in a classification-based approach.

An analysis of the performance measures of the classification-based and retrieval-based approaches shows that both of them are viable choices for implementing the Similar Program Guide. The choice of the approach for the implementation is determined by whether we need similar documents based on content or category. In order to retrieve documents similar in content, we use the retrieval-based approach, and in order to retrieve documents belonging to the same category, we use the classification-based approach. The retrieval-based approach proves to be a better option when distinction between categories is very subtle. For example, in our experiments the classification of transcripts into action or adventure was difficult. However, classification can be a good option if the transcripts can be easily classified into distinct categories.

A review of literature indicates that some of the work in the area of Spoken Data Retrieval (SDR) shares a similarity with our work. However, approaches in SDR use automated speech analysis algorithms resulting in a larger percentage of errors. Errors within closed caption transcripts are low especially when they are placed on the signal □off line□[1]. This difference makes it easy to apply intelligent information retrieval techniques more effectively to closed captions than to spoken documents generated through automated speech recognition.

In this work, we have mainly focused on evaluating some existing algorithms in light of some of the new issues underlying closed caption data, but have not proposed any new approaches. One of the directions we are pursuing is to evaluate the hypothesis that word sense disambiguation might enhance the precision of classification and retrieval for this kind of data. Word sense disambiguation is a precision-enhancing operation. Although past results in applying word sense disambiguation by others have, however, been somewhat mixed [7], we hope that the nature of closed caption data is such that the results might be more interesting.

In Section 2, we have outlined the possibility of capturing and analyzing captions from different channels to identify programs similar to a given program. Toward this end, we are also experimenting with multiplexing different channels in real time, and hope that such a technique might prove practical.

Acknowledgements. The authors acknowledge support for this work from the Georgia Electronic Design Center (formerly Yamacraw).

References

1. Dasigi V., Liu X.: Issues in automatic Classification and Retrieval of Closed Caption transcripts. In: Prasad, B. (Ed.), Proceedings of First Indian International Conference on Artificial Intelligence (2003) 521-528
2. Shardanand U. and Maes P.: Social information filtering: algorithms for automating word of mouth. In: Proceedings on Human Factors in Computing Systems (1995) 210–217
3. Balabanovic M. and Sholam Y.: Combining Content-Based and Collaborative Recommendation. Communications of the ACM, Vol. 40, No. 3 (1997) 66-72
4. McCallum, Andrew Kachites.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, http://www.cs.cmu.edu/~mccallum/bow (1996)
5. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol.34, No.1 (March 2002) 1-47
6. Dasigi, V., R. Mann, and Protopopescu. V.: Information Fusion for Text Classification - An Experimental Comparison. Pattern Recognition journal, 34 (2001) 2413-2425
7. Voorhees, E. M.: Using WordNet to disambiguate word senses for text retrieval. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (1993) 171-180

A Location Information System Based on Real-Time Probabilistic Position Inference

Takayuki Ito[1], Kazuhisa Oguri[2], and Tokuro Matsuo[1]

[1] Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso, Showa-ku, Nagoya 466-8555, Japan
{itota, tmatsuo}@ics.nitech.ac.jp
[2] Hitachi Information Systems, Ltd.
1-16-5, Dogenzaka, Shibuya-ku, Tokyo 150-8540, Japan

Abstract. In this paper, we present an implementation of a real-time position inference system for a research group. In a research group, knowing the other members' presence, predicting visitors, and holding real-time meetings are time-consuming but important tasks in everyday life. Thus, in this system, we realize a prediction mechanism for group members and visitors. We also develop a support mechanism for holding real-time meetings. In order to implement a real-time position inference system, we need to detect location information of group members. In our school building, EIRIS (ELPAS InfraRed Identification and Search System) has been installed, and we utilize EIRIS to detect members' positions. However, there exists a trade-off between the value and cost of detecting exact positions. Thus, we first try to complement EIRIS's detection ability by employing a stochastic method, probabilistic reasoning, and heuristic rules. Then, we implement a visitor prediction mechanism and a real-time meeting support mechanism. Our experiments demonstrate that our position inference mechanism is more effective than a simple prediction mechanism that employs only stochastic data.

1 Introduction

In this paper, we present an implementation of a real-time position inference system for a research group. In a research group, there are certain time-consuming but important tasks in everyday life. For example, knowing the presence and availability of group members is important for submitting reports, asking them to perform tasks, and carrying out other duties. It is also important to be aware that somebody is coming to our room and who that person is. For example, when my supervisor is coming to my room, I need to prepare for a discussion or other work.

In order to implement a real-time position inference system, we need to detect location information of group members. Many location information detection systems have been applied in the real world. The most popular one is GPS (Global Positioning System)[5]. GPS has been employed in car navigation systems since it provides effective sensing ability outside of buildings. However, we cannot employ GPSs to detect location information inside of buildings. Thus, several location detection systems

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 797-806, 2004.

have been developed for inside of buildings, e.g. Active Badge System [11], the Bat system (or the Sentient computing system)[1][4], RADAR[2], Cricket System[9], etc. In Section 5, we present the details of these systems and the difference between our system and these systems. In our school building, EIRIS (ELPAS InfraRed Identification and Search System) has been installed. EIRIS is a location information detection system made by the ELPAS (Electro-Optical Systems) Corporation. In EIRIS, each member has a badge as shown in the left of Fig. 1. Every 4 seconds, a badge sends an infrared signal to the readers shown in the right of Fig. 1. Each reader has been installed in the ceiling of the building. Our building has 8 floors. On each floor, there are several rooms, and about 20 readers have been installed.

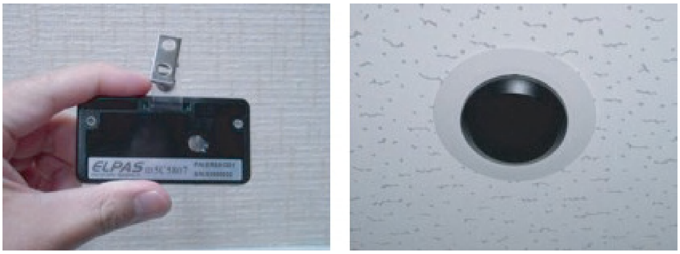


Fig. 1. Badge and Reader

In our group, by utilizing EIRIS, we have built a real-time position inference system. In this system, we realize the following two mechanisms: (1) a mechanism for predicting the presence and availability of members, and (2) a mechanism for alerting that somebody is coming to the user's office and who is coming.

In order to develop the mechanism (1) for predicting the presence and availability of a certain member, we utilize EIRIS's location detection function. However, EIRIS often fails to detect a member's position when the member is not wearing the badge in the correct position. Namely, when there is an obstacle between a reader and a badge, the reader often fails to detect a signal from the badge. If we install many readers on the walls, ceilings, etc. so that a reader can easily detect the badge's signals, we can avoid this problem. However, the cost of installing so many readers can be expensive. Even if we select another system, the cost can be even more expensive if we install a high-quality detection system. Thus, there exists a trade-off between the value and cost of detecting exact positions. This is a common problem of position detecting system. Thus, we first tried to complement EIRIS's detecting ability by employing a stochastic method, probabilistic reasoning [8], and heuristic rules.

In mechanism (1), we can gain current location information and prediction for a “one-step future”, i.e. member's next position in advance. Thus, to realize mechanism (2), we need to forecast “several-steps futures”, i.e. member's destination in advance. Thus, we propose a recursive probabilistic reasoning method in which we call mechanism (1) recursively.

This paper is organized as follows. In Section 2, we present a real-time position inference system and its main functions, i.e. predicting the presence and availability of group members and predicting visitors. In Section 3, we propose a real-time position inference mechanism based on probabilistic reasoning. In Section 4, we present the results of our experiments to show how our prediction mechanism can work in the real world. In Section 5, we compare our work with related works. Finally, we summarize our results.

2 A Real-Time Location Information System for a Research Group

The left figure of Fig. 2 shows an outline of our system. Each user wears a badge, and each badge sends infrared signals to a reader. Information from readers is first stored in the EIRIS server. Then, information is sent to our position inference server. In this server, based on the gathered information, position information is inferred in real time. The sensed information and the inferred information are shown on displays on the user's PC.

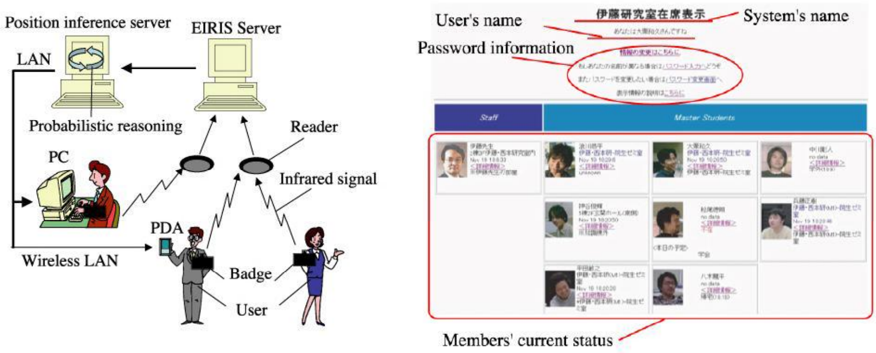
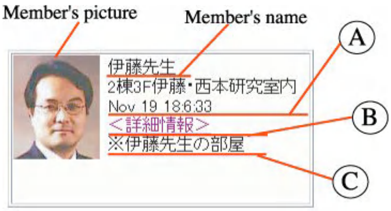


Fig. 2. System Outline and Web-based Interface

We provide two types of graphical user interfaces for the system. One is a web-based interface, and the other is an application-based interface (we omit it due to the space limitation). The right image of Fig. 2 shows the web-based interface, which has been implemented as a CGI. Users can see the status of a member via a web browser. The web page is automatically reloaded every 60 seconds. Fig. 3 shows the details of the web-based interface. Each member's current status is shown in a box. The left side of Fig. 3 shows the member's picture. The right side of Fig. 3 shows member's name, his detected position, a hyper link to his position history, and his inferred position. In the “A” of Fig. 3, detected member's position is shown. If EIRIS detects a position, that position is shown. If not, the latest position is shown. Here, the sentence means ``prof. I and prof. N's laboratory on the 3rd floor of Building No.2.” In the “B” of Fig. 3, when a user clicks this hyper link, a new window appears. The window shows histories of detected positions and inferred positions. In the “C” of Fig. 3, inferred member's position is shown. Here, the sentence means “professor I's room.”



伊藤先生
2棟3F伊藤・西本研究室内
Nov 19 18:6:33
<詳細情報>
※伊藤先生の部屋

Fig. 3. Details of Web-based Interface

3 Real-Time Position Inference based on Probabilistic Reasoning

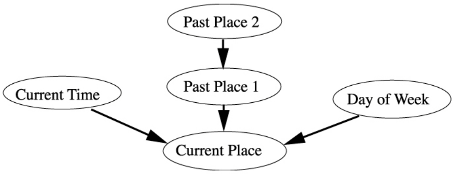
3.1 Inferring Current Position

As mentioned in Section 1, EIRIS often fails to detect a position. If EIRIS fails to detect the current position, we infer the current position based on the following three cases: (case 1) If EIRIS can detect the two most recent positions before the current position, we use a method based on a Bayesian network. (case 2) If EIRIS fails to detect positions within a certain period, i.e. 3 minutes, we stochastically infer the current position based on training data. Training data is the position history for each member. In an experiment, we gather training data over a period of about one month. (case 3) If EIRIS fails to detect positions within a certain period and the last position EIRIS detected is next of rooms that do not have a reader, we employ heuristic rules.

The details of the above three cases are given below:

(case 1): To infer the current position based on the two most recent positions, we construct a simple Bayesian network as shown in Fig. 4. Also, we employ the current time and the current day of the week to increase the accuracy of inferences. The reason why we use ``two" past positions is that if we know them, we can judge the direction of the member's movement.

Here, we consider a graph that consists of nodes and links. Each node means a reader. Each link means a movement between readers. For example, suppose that the past position data are “3-9” (indicating reader 3-9) and “3-7.” Also, the current time is 11:35, i.e. between 10:00 - 12:00, the current day of the week is Monday. Here, our system calculates the current position p that has the highest possibility based on the equation that displayed in the right of Fig. 4, where, $S=\{3\text{-}7, 3\text{-}8, 3\text{-}9, 3\text{-}10\}$.



$$\begin{aligned} p \;=\; & \arg\max_{s \in S} P(CurrentPlace = s| \\ & PastPlace1 = 3-9, PastPlace2 = 3-7, \\ & CurrentTime = 10-12, \\ & DayOfWeek = mon), \end{aligned}$$

Fig. 4. Simple Bayesian Network & Calculating current position

(case 2): If EIRIS fails to detect the past two position data, our system infers the current position based on stored data from the past. We call this stored data ``training data." In this case, our system calculates the probability of a member's existence at the place in the current time and day of the week based on the training data.
(case 3): In our school, there are several rooms where a reader is not installed. Thus, if a member's last detected position is next to a room that does not have a reader, and the member has not been detected by EIRIS, we can heuristically infer that the member is in that room. For example, the professor's room does not have a reader. Thus, if professor's last detected position is the room next to his, and the professor has not been detected by EIRIS, then we can heuristically infer that the professor has been in his room. In our school, there are several such places, e.g. professor's room, bath rooms, and refresh rooms. Thus, we introduce heuristic rules. For example, "If a member's last position is room K32, and he has not been detected in one hour, then he is in the professor's room." We created about 80 such rules for our building.

3.2 Reasoning Visitor's Destination

In order to predict coming visitors, we need to reason a member's destination. Namely, we infer a member's position in several future steps. To carry out such reasoning, we employ a Bayesian network. Here, we consider a graph that consists of nodes and links. Each node presents a reader. Each link shows a movement between readers. However, in this case, a graph would be a cyclic graph. For calculating probability in Bayesian network, we cannot have a cyclic graph [8][10]. Namely, we cannot directly employ the existing methods for calculating probability for a Bayesian network. Thus, we recursively use a reasoning method proposed in the previous section. Fig. 5 shows an example of the recursive reasoning used in our system.

First, we infer the visitor's next position based on his current position and his past position. Then, the position with the highest probability is selected as the next position. For example, in Fig. 5, based on the current place "3-9" and the past place "3-7", we can infer that the place "3-10" has the highest probability. Next, based on the visitor's current position and the already inferred next position, we infer his or her subsequent next position. In our system, we conduct several steps of reasoning to predict an approaching visitor for an user until visitor's inferred position is the user's room or the next to the user's room that does not have the reader.

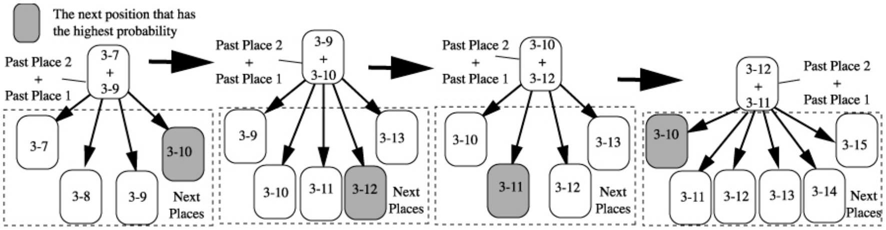

The next position that has the highest probability
Past Place 2 + Past Place 1
3-7 + 3-9
3-7
3-8
3-9
3-10
Next Places
Past Place 2 + Past Place 1
3-9 + 3-10
3-9
3-10
3-11
3-12
3-13
Next Places
Past Place 2 + Past Place 1
3-10 + 3-12
3-10
3-11
3-12
3-13
Next Places
3-12 + 3-11
Past Place 2 + Past Place 1
3-10
3-11
3-12
3-13
3-14
3-15
Next Places

Fig. 5. Recursive probabilistic reasoning

4 Experimental Results

In the experiments, we focus on the following two points.

- Accuracy of predictions by the proposed mechanism (quantitative evaluation).
- Effects of sharing position data in a group (qualitative evaluation).

In the experiments, subjects are 7 graduate students and an associate professor in a university laboratory. This laboratory's layout mainly consists of one professor's room, two students' room, and a discussion room. The students' rooms and the discussion each room have at least one EIRIS's reader, but, no EIRIS's reader has not been installed in the professor's room.

We conducted the experiment for about 2. Also, we gathered sensing data of 3 subjects ("a", "d", and "g" in the graphs A, B, and C in Fig. 6) for about 2 months as training data. From the other 4 subjects ("b", "c", "e", and "f" the graphs A, B, and C of Fig. 6), we gathered sensing data for about 10 days. Each subject could access the system via his or her own laptop computer or desktop computer. Furthermore, we provided two shared computers in the two students' rooms. At the beginning of this experiment, we explained how to use our system.

In order to clarify the accuracy of predictions by the proposed mechanism (quantitative evaluation), we conducted interviews on the actual position of each subject. A subject is randomly selected on a randomly selected date for the interview.

In the second week, we stored logs of the predicted data in our system. We compared the logs with the actual position data provided by subjects via the interview. We selected two hours on a certain date. During the two hours, the logs of the predicted data were gathered every 20 minutes.

In order to gain the actual data, we conducted interviews after the above two-hour term. At the beginning, subjects did not know when this interview would be conducted. In the interview, we asked a subject the following three questions for every log. Actually, based on these questions, we could obtain the actual position data of subjects during the experimental period. (Q.1) "Was there any activity that was not sensed or not predicted by our system?" (Q.2) "During this time, did you wear your badge?" Through this question, we could find out know when the subject did not wear a badge. If our system logged the subject's position when he did not wear a badge, we could find out that he placed his badge on his desk, etc. (Q.3) ``During this time, when you did not wear your badge, where were you?"

In this experimentation, we do not consider day of the week in probabilistic networks for the following reasons. First, the length of the experimentation term was not long. If we considered day of the week, the number of data would be quite small. Second, in the term of the experimentation, there were no scheduled lectures. Thus, there were no week-based activities for students. As a future work, we need to include information on day of the week, part of the month, month of the year, etc. Here, we concentrate on primitive information, i.e. subjects' past position data and the subjects' schedules that are explicitly indicated by them.

In order to analyze the effects of sharing position data in a group (qualitative evaluation), we conducted two questionnaires. In one questionnaire, a subject could answer questions at anytime. Each subject answered the other questionnaire after the term of this experimentation.

4.1 Accuracy of Predicted Data

In this section, we compare the predicted data and the actual history of subjects' behavior. First, we conducted the following two comparisons: [Comparison 1] Comparison of all methods (heuristic rules, probabilistic reasoning method, and stochastic inference method) in our system with real data (actual position data). [Comparison 2] Comparison of only stochastic inference method in our system with real data (actual position data). In these comparisons, we clarify the effects of heuristic rules and the probabilistic reasoning method. The graph A of Fig. 6 shows the results, where the vertical axis indicates accuracy and the horizontal axis indicates data numbers. There are 13 data gathered by 7 subjects. For example, 1.a and 2.a are gathered at different times from subject a.

In almost all cases, the accuracy of comparison 1 is higher than the accuracy of comparison 2. This result is quite natural, since in comparison 1 all methods are employed. In two exceptions, i.e., data 9.e and 6.c, the accuracy of comparison 2 was higher than that of comparison 1. In particular, in 9.c, the difference is large. The following is the reason for this large difference. When EIRIS cannot detect the position, and the last position is different from the current position, there is a possibility that the predicted current position will be different from the actual current position. If there is not enough training data for predicting the current position from the last position, the prediction can be incorrect. Especially for subject e, the amount of training data is not sufficient for prediction. Thus, the accuracy of comparison 2 is higher than that of comparison 1.

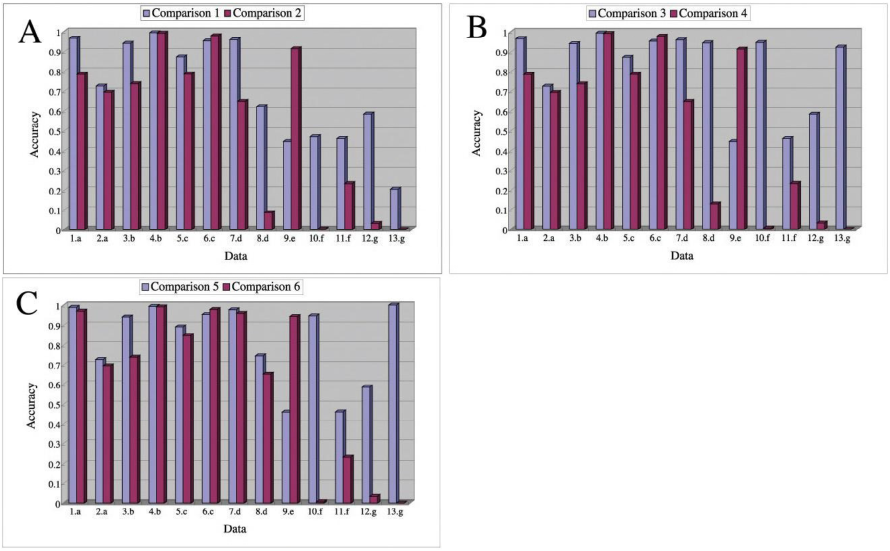


Fig. 6. The results of Comparisons

For subjects f and g, the accuracy based on comparison 2 is extremely small. This is because subject f wore his badge only a short time, and subject g's room does not have an EIRIS reader. Thus, the stochastic method could not work correctly, and the heuristic rules could work well.

While the average accuracy of comparison 1 is 0.707, the average accuracy of comparison 2 is 0.529. Clearly, when we employ all methods (heuristic rules, probabilistic reasoning method, and stochastic inference method), the accuracy is higher than when we employ only the stochastic inference method.

The above results for comparison 1 and comparison 2 include data that are gathered when the subject is out of the school building. EIRIS cannot detect positions outside of the building. Thus, secondly, we made two more comparisons that focused on the time when each subject is inside of the building. [Comparison 3] Comparison of all methods (heuristic rules, probabilistic reasoning method, and stochastic inference method) in our system with real data during the time when the subject is in the school building. [Comparison 4] Comparison of only stochastic inference method in our system with real data during the time when the subject is in the school building.

The graph B of Fig. 6 shows the accuracies of comparison 3 and comparison 4. While the average accuracy of comparison 3 is 0.825, the average accuracy of comparison 4 is 0.533. The accuracies of comparison 3 and comparison 4 are higher than those of comparison 1 and comparison 2. This is because we focus on the time spent inside of the building.

Furthermore, we conducted two more comparisons that focused on the places where EIRIS readers are installed. For example, the professors' rooms have no EIRIS readers. [Comparison 5] Comparison of all methods (heuristic rules, probabilistic reasoning method, and stochastic inference method) in our system with real data when the subject is in a place where the reader is installed. [Comparison 6] Comparison of only stochastic inference method in our system with real data when the subject is in the place where a reader is installed.

The graph C of Fig. 6 shows the results of comparison 4 and comparison 5. Here, the accuracy of comparison 5 is higher than that of comparison 1. In addition, the accuracy of comparison 6 is higher than those of comparison 2 and comparison 4.

In the above experiment, we could achieve sufficient results to show the effectiveness of our system's inference mechanism. One reason for this high effectiveness is that subjects were almost always moving and acting based on their usual patterns in the building.

5 Related Work

In this section, we compare our work with related works. Active Badge System [11] is an indoor-location sensing system that utilizes infrared systems. The Bat system (or the Sentient computing system) [1][4] is a context-aware system that consists of mobile/fixed wireless transmitters, receiver elements, and a central radio frequency base station. RADAR [2] is an indoor positioning system that implements a location service based on a radio frequency data network. The Cricket [9] is a supersonic-based positioning system. Unlike our system, these systems do not focus on predicting user.

In the field of probabilistic reasoning, there have been several studies on prediction based on Bayesian networks. In the BATMobile project [3], an automated car driving method was proposed based on a dynamic Bayesian network (DBN) [10]. A DBN could be applied to our system, but we do not employ a DBN since the cost for calculation can be quite huge. The Coordinate [6] supports collaboration by learning predictive models that provide forecasts of users' presence and availability by using Bayesian networks. In Coordinate, the main issue is predicting the presence and availability of users based on stored data. This work did not focus on position detection systems and did not consider that they often fail to detect positions. Our system tries to show the real-time positions by using probabilistic reasoning and heuristic rules even if the position detection system fails to detect the users' positions. Also, probabilistic reasoning has been applied to object identification problems [7]. In these problems, the goal was to decide if some newly observed object was the same as some previously observed object. In the traffic surveillance application studied in these works, the sensors are cameras at several positions on a freeway network. The objects are cars that do not have any specific device. On the contrary, in our system, a member has a badge, and we try to use reasoning to determine the member's positions.

6 Conclusions

In this paper, we presented an implementation of a real-time position inference system for a research group. In our system, we developed a prediction mechanism for members' and visitors' presence. We utilized EIRIS, which has been installed in our graduate school for detecting members' positions. However, there exists a trade-off between the value and cost of detecting exact positions. Thus, we first complemented EIRIS's detection ability by employing a stochastic method, probabilistic reasoning, and heuristic rules. Then, based on this approach, we implemented a visitor prediction mechanism and a real-time meeting support mechanism. Our experiments demonstrated that our position inference mechanism is more effective than a simple prediction mechanism that employs only stochastic data.

References

1. M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. IEEE Computer, 34(8): 50-56, Aug 2001.
2. P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In Proc. of IEEE Inforcom 2000, pages 775 - 784, 2000.
3. J. Forbes, T. Huang, K. Kanazawa, and S. Russell. The BATmobile: Towards a bayesian automated taxi. In Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI95), pages 1878-1885, 1995.
4. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. Wireless Networks, 8:187-197, 2002.
5. Hofmann-Wellenhof, H. Lichatenegger, and J. Collins. Global Positioning System: Theory and Practice. Springer Verlag, 2000.

6. E. Horvitz, P. Koch, C. M. Kadie, and A. Jacobs. Coordinate: Probabilistic forecasting of presence and availability. In Proc. of the 18th Conference on Uncertainty and Artificial Intelligence (UAI02), pages 224-233, 2002.
7. H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In the Proc. of 16th International Joint Conference on Artificial Intelligence (IJCAI99), 1999.
8. J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan kaufmann Publishers, Inc., 1988.
9. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In Proc. of 6th International Conference on Mobile Computing and Networking (Mobicom00), pages 32-43. ACM Press, 2000.
10. S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice-Hall, Inc., 1995.
11. R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. ACM Transactions on Information Systems, pages 91-102, Jan. 1992.

Expertise in a Hybrid Diagnostic-Recommendation System for SMEs: A Successful Real-Life Application

Sylvain Delisle[1] and Jos□e St-Pierre[2]

Institut de recherche sur les PME
Laboratoire de recherche sur la performance des entreprises
Universit□du Qu□bec □Trois Rivi□res
1: D□partement de math□matiques et d□informatique
2: D□partement des sciences de la gestion
C.P. 500, Trois-Rivi□res, Qu□bec, Canada, G9A 5H7
{sylvain_delisle, josee_st-pierre}@uqtr.ca
www.uqtr.ca/{~delisle, dsge}

Abstract. We describe a hybrid expert diagnosis-recommendation system we have developed for SMEs. The system is fully implemented and operational, and has been successfully put to use on data from actual SMEs. Although the system is packed with knowledge and expertise, it was not implemented with □traditional□ symbolic AI techniques. We explain why and discuss how the system relates to expert systems, decision support systems, and AI. We also report on an experimental evaluation and identify ongoing and future developments.

1 Introduction

In this application-oriented paper, we briefly describe a hybrid expert system we have developed for SMEs. Using a benchmarking approach [1, 23], our system performs a multidimensional evaluation of a SME□s production and management activities, and assesses the results of these activities in terms of productivity, profitability, vulnerability and efficiency. This system is fully implemented and operational, and has been put to use on actual data from some 500 SMEs from Canada, USA, and France. By academic standards, it is clearly a successful real-life application.

What is peculiar though, especially from a knowledge-based systems perspective, is the fact that although the PDG system is packed with knowledge and expertise on SMEs, it has not been implemented with □traditional□symbolic Artificial Intelligence (**AI**) techniques. However, the PDG system certainly qualifies as a hybrid diagnostic-recommendation expert system. In the following sections, we provide further details on the PDG system and how it relates to knowledge-based and expert systems, as well as to decision support systems. We also identify aspects of the PDG that will benefit from the addition of AI techniques in current and future developments.

Our work takes place within the context of the Research Institute for SMEs. The Institute□s core mission is to support fundamental and applied research to foster the advancement of knowledge on small and medium-sized enterprises (**SMEs**) to con-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 807-816, 2004.

tribute to their development. Our lab, the LaRePE *(LAboratoire de REcherche sur la Performance des Entreprises*: `www.uqtr.ca/inrpme/larepe/`), is mainly concerned with the development of scientific expertise on the study and modeling of SMEs□performance, including a variety of interrelated subjects such as finance, management, information systems, production, technology, etc. All research projects carried out at the LaRePE involve both theoretical and practical aspects, always attempting to provide practical solutions to real problems confronting SMEs, often necessitating in-field studies.

2 Expert Systems, Decision Support Systems, and AI

Since results produced by the PDG system can be used by SME owners to make decisions regarding the management of their business, the PDG system can be regarded as a decision support system (**DSS**) [16, 20] that uses benchmark data to compare SMEs against their peers□ a group of peers constitutes a reference group for benchmarking purposes. We now look at various aspects of expert systems (**ES**), DSS, and AI that are relevant to the conceptual background of the PDG system before we get into the specifics of that system in Section 3.

DSSs exist to assist someone in making a choice, rendering a judgement, or drawing a conclusion. Their operation is subordinated to the human user, who remains central to and in control of the decision-making process. Because ESs however emulate the human reasoning process, they typically operate automatically until they reach a conclusion which is then communicated to their user. This observation is supported by Forgionne *et al.* [3], who adopt the same fundamental division of decision support mechanisms into autonomous or assistive systems. ESs and DSSs are applied, driven by exigencies encountered in the real world, rather than a working-out of some conception of a theoretical model.

Although ESs and other AI techniques are often used in DSSs, we are unaware of any implementation in which a DSS assists an ES save that proposed by Plenert [11], which would use a DSS for a period of time to capture decisions which could subsequently be codified to produce the rules of an ES. A quite substantial literature on the other hand discusses the use of AI in DSS. While it would be generally agreed that symbolic ES approaches are the AI techniques used most often in DSSs, it should be noted that non-symbolic decision-making methodologies such as neural nets [5] and genetic algorithms [22] are also employed.

Houben *et al.* [4] argue for the use of a knowledge-based DSS to make strategic decisions in DSSs. Although their system for identifying and assessing organizational strengths, weaknesses, opportunities and threats (□SWOT□) appears not to have actually been put to use, the authors report that the □general opinion of the experts about the validity of the system was quite positive and encouraging□and that □only about 10% of the output of the system was doubted by the experts□ Rosca and Wild [13] survey the study and use of business rules, which are natural language statements expressing the policies of an enterprise in a manner akin to the antecedent-consequent syntax commonly used in an expert system. Business rules thus separate out the rulebase from the ES and treat it as an object of study in itself.

Only limited research has been done into the particular area that interests us, the use of ESs in DSSs which benchmark small and medium-sized industrial enterprises. Muhlemann *et al.* [9] report on the adoption and evolution of a generic production management DSS in two SMEs. They found that the system acted as a □change agent□, evolving to encompass decisions of greater scope, and that this evolution was facilitated by the good production management practices embodied in the software. Price *et al.* [12] describe a DSS for manufacturing enterprises which has the goal of supporting more frequent changes in corporate strategy, an emerging business requirement the authors identify. The nature and number of DSS inputs approximate those involved in benchmarking efforts.

Levy and Powell [6] study strategies for SME information systems. One of the main benefits identified of adopting strategic information systems is □obtaining information to manage the business more effectively and competitively□, the goal of DSSs. The same authors [7] also found that although many of the lessons learned in large firms remain valid for small ones, there are a number of key differences, such as a dependence on external environment and the absence of formal information system departments in SMEs. These differences are pertinent to DSSs because they presume the existence in the organization of an information system able to produce operational data on which decisions are based.

3 SME Performance Evaluation with the PDG System

3.1 An Overview of the System

The PDG system, which runs on computers located in our lab, evaluates a SME from an external perspective and on a comparative basis in order to produce a diagnosis of its performance and potential, complemented with relevant recommendations□ the project started in 1997 and the current version has been in production for 2 years. The PDG system is a hybrid diagnostic-recommendation system as it not only identifies the evaluated SME□s weaknesses, but it also makes suggestions on how to address these weaknesses in order to improve the SME□s performance. An extensive (18-page) questionnaire is used to collect relevant information items on the SME to be evaluated, along with the financial statements of the last five years. Data extracted from the questionnaire and the financial statements is computerized and fed into the system. The latter performs a multidimensional evaluation in approximately 3 minutes by contrasting the particular SME with an appropriate reference group of SMEs for which we have already collected relevant data; this is the crux of the benchmarking process. The output is a detailed report in which 28 management practices (concerning human resources management, production systems and organization, market development activities, accounting, finance and control tools), 20 results indicators and 22 general information items are evaluated, leading to 14 recommendations on short term actions the evaluated SME could undertake to improve its overall performance□ details of the main computations involved during benchmarking are presented in [18]. For instance, here is an example recommendation on human resources results: *"Your overall effectiveness at managing human resources is comparable to that of* *your reference group. You should pay attention to why certain managerial jobs have a high rate of voluntary departure, with the objective of lowering hiring and training costs".*

As shown in Figure 1, the PDG expert diagnosis system is connected to an Oracle database which contains all the relevant data for benchmarking purposes. The PDG reports are constantly monitored by a team of multidisciplinary human experts in order to ensure that recommendations are valuable for the entrepreneurs. This validation phase, which always takes place before the report is sent to the SME, is an occasion to make further improvements to the PDG system, whenever appropriate. It is also a valuable means for the human experts to update their own expertise on SMEs. Figure 1 also shows that an intermediary partner is part of the process in order to guarantee confidentiality: nobody in our lab knows to what companies the data are associated.

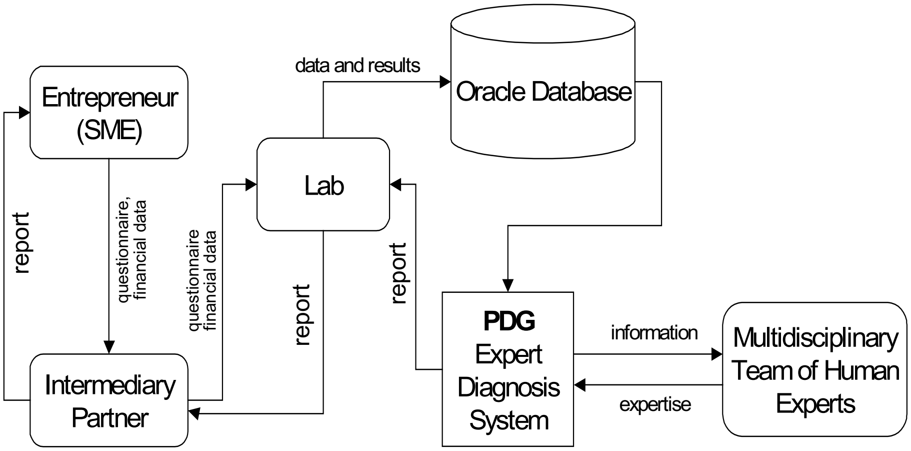

data and results
Oracle Database
Entrepreneur (SME)
Lab
report
questionnaire, financial data
questionnaire financial data
report
report
PDG Expert Diagnosis System
information
expertise
Multidisciplinary Team of Human Experts
Intermediary Partner

Fig. 1. The PDG system: evaluation of SMEs, from an external perspective and on a comparative basis, in order to produce a diagnosis of their performance and potential

As far as we know, our system is unique. A somewhat similar system is presented in [2]. However, their system was especially developed for SMEs that produce goods in relatively small volumes and in batch. It is based on a specific benchmark focusing on manufacturing and assembly processes. Moreover, the system they describe is mostly semi-automatic (if not mostly manual), whereas ours is entirely automatic, let alone a final revision of the final wording of the main recommendations which usually takes between five to ten minutes.

3.2 Some Details on the System

Besides an Oracle database, the PDG uses the SAS statistical package, and Microsoft Excel (see Figure 2 next page). The system□s expertise is located in the questionnaire and the benchmarking results interpretation module. The first version of the questionnaire was developed by a multidisciplinary team of researchers in business strategy, human resources, information systems, industrial engineering, logistics, marketing, economics, and finance. The questionnaire development team was faced with two important challenges: 1) find a common language (a shared ontology) that would allow researchers to understand each other and, at the same time, would be accessible to entrepreneurs when answering the questionnaire, and 2) identify long-term performance indicators for SMEs, as well as problem indicators, while keeping contents to a minimum since in-depth evaluation was not adequate.

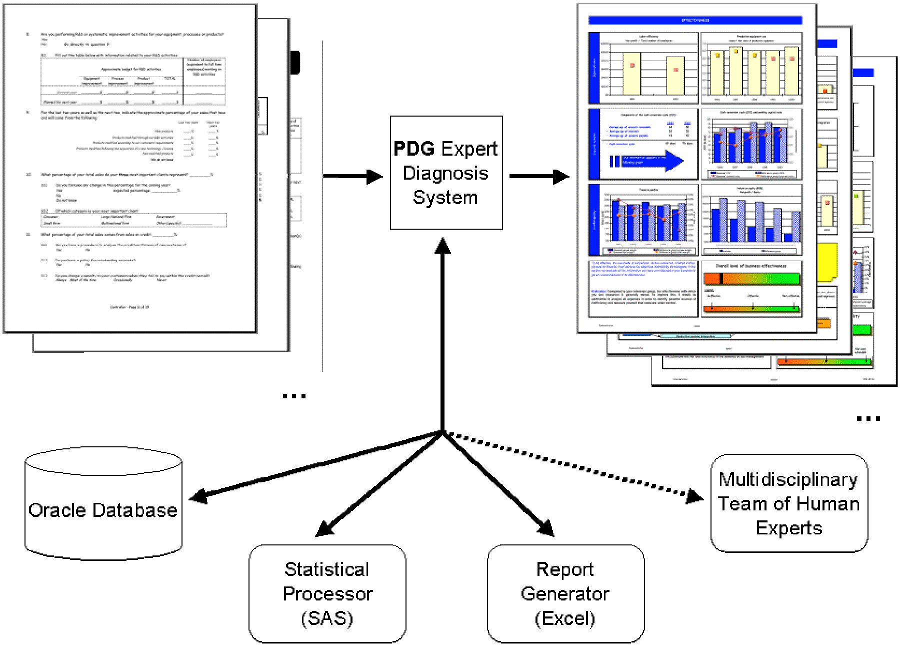

PDG Expert Diagnosis System
...
...
Oracle Database
Statistical Processor (SAS)
Report Generator (Excel)
Multidisciplinary Team of Human Experts

Fig. 2. The PDG system and its main components: from a broad-coverage questionnaire and financial statements to a detailed benchmarking evaluation and recommendations.

The team was able to meet these two goals by the assignment of a □knowledge integrator□role to the project leader. During the 15-month period of its development, the questionnaire was tested with entrepreneurs in order to ensure that it was easy to understand both in terms of contents and question formulation, and report layout and information visualization. All texts were written with a clear pedagogical emphasis since the subject matter was not all that trivial and the intended readership was quite varied and heterogeneous. Several prototypes were presented to entrepreneurs and they showed a marked interest for graphics and colours.

The researchers□expertise was precious in the identification of vital information that would allow the PDG system to rapidly produce a general diagnosis of any manufacturing SME. The diagnosis also needed to be reliable and complete, while being comprehensible by typical entrepreneurs. This was pioneering research work that the whole team was conducting. Indeed, other SME diagnosis systems are generally financial and based on valid quantitative data□ see [8, 10, 21] for examples of expert systems in finance. Each expert had to identify practices, systems, or tools that had to be implemented in a manufacturing SME to ensure a certain level of performance. Then, performance indicators had to be defined in order to measure to what extent these individual practices, systems, or tools were correctly implemented and allowed the enterprise to meet specific goals□ the relationship between practices and results is a distinguishing characteristic of the PDG system. Next, every selected performance indicator was assigned a relative weight by the expert and the knowledge integrator. This weight is used to position the enterprise being diagnosed with regard to its reference group, thus allowing the production of relevant comments and recommendations. The weight is also used to produce a global evaluation that will be displayed in a synoptic table. Contrary to many performance diagnostic tools in which the enterprise□s information is compared to norms and standards (e.g. [8]), the PDG system evaluates an enterprise relative to a reference group selected by the entrepreneur. Research conducted at our institute seriously questions this use of norms and standards: it appears to be dubious for SMEs as they simply are too heterogeneous to support the definition of reliable norms and standards.

Table 1. Some aspects of the representation of expertise within the PDG system with performance indicators implemented as variables. This table shows three (3) variables: one scale variable (participative management), one binary (remuneration plan), and one continuous numerical (fabrication cost). **Legend**: SME = variable value for the evaluated enterprise; MEA = mean value of the variable in the reference group; RG = reference group; MED = median value of the variable in the reference group; CODE = resulting code for the evaluated enterprise.

Scale variable	**Binary variable**	**Continuous (numerical) variable**
if SME >= (1.25 x MEA), then CODE = 4	if SME = 1 and 10% of RG = 1 then CODE = 4	if SME >= (1.25 x MED), then CODE = 4
if SME >= (1.10 x MEA), then CODE = 3	if SME = 1 and 25% of RG = 1 then CODE = 3	if SME >= (1.10 x MED), then CODE = 3
if SME >= (1.00 x MEA), then CODE = 2	if SME = 1 and 50% of RG = 1 then CODE = 2	if SME >= (1.00 x MED), then CODE = 2
if SME >= (0.90 x MEA), then CODE = 1	if SME = 1 and 75% of RG = 1 then CODE = 1	if SME >= (0.90 x MED), then CODE = 1
if SME >= (0.75 x MEA), then CODE = 0	if SME = 1 and 90% of RG = 1 then CODE = 0	if SME >= (0.75 x MED), then CODE = 0
Example: participative management	*Example: remuneration plan*	*Example: fabrication cost*

Performance indicators are implemented as variables in the PDG system□ more precisely in its database, and in the benchmarking results interpretation module (within the report production module). These variables are defined in terms of three categories: 1) binary variables, which are associated with yes/no questions; 2) scale variables, which are associated with the relative ranking of the enterprise along a 1 to 4 or a 1 to 5 scale, depending on the question; and 3) continuous (numerical) variables, which are associated with numerical figures such as the export rate or the training budget. Since variables come in different types, they must also be processed differently at the statistical level, notably when computing the reference group used for benchmarking purposes. In order to characterize the reference group with a single value, a central tendency measure that is representative of the reference group□s set of observations is used. Depending on the variable category and its statistical distribution, means, medians, or percentages are used in the benchmarking computations. Table 1 (previous page) shows an example of how the evaluated enterprise□s results are ranked and associated with codes that will next be used to produce the various graphics in the benchmarking report.

AI appears in the PDG software primarily as an embedded ES whose rules are implemented as formulae attached in the cells of an Excel spreadsheet that composes the PDG report. It is worth noting that antecedents, *if-clauses*, in these rules are not parameterized only for client instance data, as is the practice in one class of ES. Instead, both instance data (the data evaluated) and peer group data (the evaluation context) can be parameters. Thus rules in the PDG system have the general form □if client-SME is 20% less than group_mean ...□where client-SME and group_mean are variables. Many ES consequents, *then-clauses*, use meaningful variable names and are structured so their meaning is fairly straightforward to the reader. Thus, the rule example above might have the consequent, □.. then feed_stock is incorrect□ Readable rules make it easier to implement the explanation facilities commonly found in ESs. A PDG rule consequent operates indirectly: its execution does nothing except concatenate an appropriate string from short text fragments. The content of that string is the counterpart of the traditional rule consequent, detailing a course of action to be carried out, not by the system, but by the benchmarking report reader. Finally, with respect to the explanation facility of traditional ES, one might say that when read together, the evaluation and recommendation paragraphs produced by PDG rules are self-explanatory; that is, together they provide their own explanation. Implementing an ES as cell formulae in Excel is not the most obvious way to do it, and the motivation for this fairly unusual approach is further explained in the next section.

4 Knowledge Engineering Aspects

A good deal of multi-domain expertise and informal knowledge engineering was invested into the design of the PDG system. In fact, at the early stage of the project, it was even hoped that a traditional expert-system approach would apply naturally to the task we were facing. Using the Visual Rule Studio expert system shell, a prototype was in fact developed for a subset of the full system dealing only with human resources. However, the knowledge acquisition, knowledge modelling, and knowledge validation/verification phases [14, 15] were too demanding in the context of our resources constraints, especially in the context of a multidisciplinary domain such as that of SME for which little formalized knowledge exists. Indeed, many people were involved, all of them in various specialization fields, and with various backgrounds (researchers, graduate students, research professionals, entrepreneurs). The development of a truly multidimensional performance evaluation scheme especially tailored to SMEs was, and still is, a quite demanding and challenging endeavour.

One of the main difficulties that hindered the development of the prototype expert system was the continuous change both the questionnaire and the benchmarking report were undergoing during the first three years of the project. At the same time the research team was trying to develop a multidimensional model of SME performance evaluation, users□needs had to be considered, software development had to be carried out, and evaluation reports had to be produced for participating SMEs. This turned out to be a rather complicated situation. The human-resources prototype was developed (with the expert system shell) in parallel with the full system version (which was based mainly on database technology). The project leader□s knowledge engineer role was very difficult since several experts from different domains were involved and the extraction and fusion of these various fields of expertise had never been done before. Despite the experts□valuable experience, knowledge, and good will, they had never been part of a similar project before. The modelling of such rich, complex, and vast information was an entirely new challenge both scientifically and technically. Indeed, because of their heterogeneous nature, and contrary to large enterprises, SMEs are much more difficult to model and evaluate. Finally, a comparison between the human-resources results produced by the prototype with those produced by the full system revealed that the latter were better.

These important considerations and difficulties, not mentioning the consequences they had on the project□s schedule and budget, lead to the abandon of the expert system approach in favour of the database approach. We had to make a rational decision based on our experience, but also on the well-documented fact that multi-domain, multi-expert knowledge acquisition and modelling constitutes a great challenge. Yet another factor that had great influence on our design decisions was the fact that the project started out on paper as a questionnaire, which led naturally to database building and use of all the database-related techniques and technology.

5 Experimental Evaluation

A recent experimental evaluation [17] was performed in 2002 on the inventory of reports produced to that date for 307 SMEs, 258 first-time and 49 second-time users. The authors hypothesized that the use of benchmarking would have a positive effect on enterprise operational and financial outcomes, and tested this by comparing the results involved for the first-time and second-time populations as reported in their profiles. Although not all differences are significant, non-parametric statistical analysis (t- and z-scores) showed general support for the hypothesis. Further examination of the analysis indicated that the 49 second-time users □showed marked improvement in their financial performance from the first to second year□ An attempt was then made to gain insight into the relationships between the factors in play using structural equation modelling (partial-least-squares). It suggested that benchmarking has a positive effect on operational performance (γ = .13) and financial performance (γ = .08), and tends to lead to the implementation of best practices (γ = .16). The authors explain the lower gamma for financial performance as reflecting the time required for increases in productivity to show up on the bottom line.

6 Conclusion: Toward an Even More Intelligent PDG System

We have briefly presented a fully implemented expert diagnostic system which evaluates on a benchmarking basis the performance of SMEs. The PDG system has been in use for several years and has gone through a constant and quite challenging evolution in order to meet both the needs of SME-oriented research (numerous research projects have used the system so far) and the production of benchmarking reports for hundreds of SMEs. The current version of the PDG system is not implemented with traditional AI techniques, e.g. knowledge base of symbolic rules and facts, inference engine, etc. Instead, the full system ended up as knowledge-packed system built on database technology that allows it to produce outputs that only a human expert, or in fact several human experts in different domains, would be able to produce in terms of diagnosis and recommendation quality. The output report contains mostly coloured diagrams and simple explanations that are formulated in plain English (or French) so that SMEs entrepreneurs can easily understand it. The system also uses some conventional techniques, e.g. the comments produced in the output report are generated via a template-based approach, an early technique used in natural language processing.

Our system is now at a stage where we can now consider the introduction of AI techniques in new developments. Here are three examples. 1) The huge number of database attributes and statistical variables manipulated in the system is overwhelming. A conceptual taxonomy, coupled with an elaborated data dictionary, has now become a necessary addition. This work is in progress. 2) Augmenting the PDG system with case-based reasoning seems a promising avenue. Evaluation of the problem at hand could be facilitated if it were possible to establish relationships with similar problems (cases) already solved before□ e.g. see [19]. Determining the problems□salient features to support this approach would also offer good potential to lessen the users□burden during the initial data collection phase. This is part of our future work. 3) The development of data warehouses and data mining algorithms to facilitate statistical processing of data and extend knowledge extraction capabilities is a priority. This is the main focus of our current work. Indeed, we are about to activate a new data warehouse and start our first experimentations with data mining techniques. Data warehousing has many practical uses in our SME-oriented context and it will, along with data mining techniques, positively affect the way researchers use the rich data we have collected and continue to collect on SMEs. We hope to significantly extend our knowledge on SMEs, and further improve our evaluation model of SME performance.

References

1. Cassell C., S. Nadin & M.Older Gray (2001), "The Use and Effectiveness of Benchmarking in SMEs", *Benchmarking: An International Journal*, 8(3), 212-222.
2. Denkena B., R. Apitz & C. Liedtke (2003), "Knowledge-Based Benchmarking of Production Performance", *Proc. of the First International Conf. on Performance Measures, Benchmarking and Best Practices in the New Economy (Business Excellence '03)*, Guimaraes (Portugal), 10-13 June 2003, 166-171.

3. Forgionne G., R. Kohli & D. Jennings (2002), “An AHP Analysis of Quality in AI and DSS Journals”, *International Journal of Management Science,* 30, 171-183.
4. Houben G., K. Lenie & K. Vanhoof (1999), “A Knowledge-Based SWOT-Analysis System as an Instrument for Strategic Planning in Small and Medium Sized Enterprises”, *Decision Support Systems*, 26(2), 125-135.
5. Leung Y.W. & J.-Y. Mao (2003), “Providing Embedded Proactive Task Support for Diagnostic Jobs: A Neural Network-Based Approach”, *Expert Systems with Applications*, 25(2), 255-267.
6. Levy M. & P. Powell (2000), “Information Systems Strategy for Small and Medium Sized Enterprises: An Organisational Perspective”, *The Journal of Strategic Information Systems*, 9(1), 63-84.
7. Levy M., P. Powell & R. Galliers (1999), “Assessing Information Systems Strategy Development Frameworks in SMEs”, *Information & Management*, 36(5), 247-261.
8. Matsatsinis N.F., M. Doumpos & C. Zopounidis (1997), “Knowledge Acquisition and Representation for Expert Systems in the Field of Financial Analysis”, *Experts Systems with Applications*, 12(2), 247-262.
9. Muhlemann A., D. Price & M. Afferson (1995), “A Computer Based Approach for Enhancing Manufacturing Decision Making in Smaller Manufacturing Enterprises: A Longitudinal Study”, *International Journal of Management Science,* 23(1), 97-107.
10. Nedovic L. & V. Devedzic (2002), “Expert Systems in Finance—A Cross-Section of the Field”, *Expert Systems with Applications*, 23, 49-66.
11. Plenert G. (1994), “Improved Decision Support Systems Help to Build Better Artificial Intelligence Systems”, *Kybernetes*, 23(9), 48-54.
12. Price D., R. Beach, A. Muhlemann, J. Sharp & A. Paterson (1998), “A System to Support the Enhancement of Strategic Flexibility in Manufacturing Enterprises”, *European Journal of Operational Research*, 109, 362-376.
13. Rosca R. & C. Wild (2002), “Towards a Flexible Deployment of Business Rules”, *Expert Systems with Applications*, 23, 385–394.
14. Santos J., Z. Vale & C. Ramos (2002), “On the Verification of an Expert System: Practical Issues”, *Lecture Notes in Artificial Intelligence #2358*, 414-424.
15. Schreiber G., H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de velde & B. Wielinga (2002), *Knowledge Engineering and Management: The Common KADS Methodology*, MIT Press.
16. Shim J.P., M. Warkentin, J.F. Courtney, D.J. Power, R. Sharda & C. Carlsson (2002), “Past, Present, and Future of Decision Support Technology”, *Decision Support Systems*, 33, 111-126.
17. St-Pierre J., L. Raymond & E. Andriambeloson (2002), “Performance Effects of the Adoption of Benchmarking and Best Practices in Manufacturing SMEs”, *Proc. of the Conf. on Small Business and Enterprise Development*, The University of Nottingham (UK), 15-16 April 2002.
18. St-Pierre J. & S. Delisle (2004), “An Expert Diagnosis System for the Benchmarking of SMEs’ Performance”, *Benchmarking—An International Journal*, Emerald, 11(5-6), to appear.
19. Stamelos I. & I. Refanidis (2002), “Decision Making Based on Past Problem Cases”, *Lecture Notes in Artificial Intelligence #2308*, 42-53.
20. Turban E. & J.E. Aronson (2001), *Decision Support Systems and Intelligent Systems*, Prentice Hall.
21. Wagner W.P., J. Otto & Q.B. Chung (2002), “Knowledge Acquisition for Expert Systems in Accounting and Financial Problem Solving”, *Knowledge-Based Systems*, 15, 439-447.
22. Wong M.L. (2001), “A Flexible Knowledge Discovery System Using Genetic Programming and Logic Grammars”, *Decision Support Systems,* 31(4), 405-428.
23. Yasin M.M. (2002), “The Theory and Practice of Benchmarking: Then and Now”, *Benchmarking: An International Journal*, 9(3), 217-243.

How to Speed Up Reasoning in a System with Uncertainty?

Beata Jankowska

Poznań University of Technology,
Institute of Control and Information Engineering,
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
Beata.Jankowska@put.poznan.pl

Abstract. Each expert system provides its users with a great amount of knowledge of a domain. Moreover, a friendly system-user interface usually guarantees an easy access to this knowledge. However, an expert system is sometimes too slow to serve its purpose.
Most of expert systems are being implemented as rule-based systems with uncertainty (uncertain knowledge, inexact reasoning). There are a few methods of certainty factors calculation. The well known Dempster's rule of combination always gives a result which is precise and independent of a strategy of conflict resolution. In this paper, we propose a new method, called the rule of convergence. Giving an approximate final result, it can significantly speed up the whole process of reasoning.
The methods of uncertain knowledge representation and the methods of inexact reasoning are willingly implemented in medical expert systems. The rule of convergence can really raise their standard of efficiency.

Keywords: expert systems, inexact reasoning, medical prognosticating.

1 Introduction

Considering lots of information being conveyed at present, we can really appreciate expert systems and their efficiency. An expert system provides the user with pre-processed knowledge of a domain. Such knowledge can be applied immediately. An expert system communicates with its user by means of a simple, not specialised, semi-natural language.

Unfortunately, an expert system is not devoid of drawbacks. It often fails to fulfil our expectations. At the worst, it may even work incorrectly. This may be either due to its wrong exploitation or □frequently - to its defective construction. The latter one may be a consequence of inserting wrong facts or dependencies into the system knowledge base. An expert system can also disappoint because of drawing only uncertain (not certainly known!) conclusions from the facts in its knowledge base. Last but not least, an expert system is sometimes too slow to serve its purpose.

Medicine is one of the most important and interesting domains of the expert systems application. A doctor, specialist in some branch of medicine, is unable to diagnose most of the illnesses which are not related with the field of his or her

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 817-826, 2004.

interest. In such situations, choosing a patient-specific pharmacotherapy is a yet more difficult task. Furthermore, doctors often work under strong pressure of time. They do have to take their decisions immediately!

2 Uncertainty and Its Implementations

Generally speaking, what we mean by the term uncertainty is lack of information needed to make a decision. The uncertainty is the subject of many formal theories, among others:

- Pascal-Fermat theory, introduced in the 18th century and considered to be the classical theory of probability;
- Carnap theory, announced in 1950, that pointed at the new type of probability, so called epsitemic probability;
- Dempster-Shaffer theory, being developed in the sixties and seventies of the 20th century, according to the spirit of Carnap theory.

What makes the main difference between Pascal-Fermat and the two remaining theories is the way how an ignorance notion is used. Namely, the classical theory claims that the evidence not supporting a hypothesis H is an evidence for the refutation of H. There is no place for ignorance here (follows from the axiom: $P(H) + P(H') = 1$). Unlike the first theory, in the remaining ones, while lacking the knowledge about H, we do not have to assign any belief to H or to its negation H'. Instead, we may assign the remaining belief to the environment (the set of all possible hypotheses) Θ.

To realize this idea, Dempster and Shaffer consider the following factors in their theory:

- the certainty factor cf: $2^\theta \rightarrow [0; 1]$, fulfilling the conditions:

$$cf(\varnothing) = 0 ,$$
$$\Sigma(X \in 2^\theta)\ cf(X) = 1 ;$$

- the global certainty factor Bel: $2^\theta \rightarrow [0; 1]$, such that:

$$Bel(H) = \Sigma(X \subseteq H)\ cf(X) ;$$

- the refutation factor Dbt: $2^\theta \rightarrow [0; 1]$, such that:

$$Dbt(H) = Bel(H') ;$$

- the plausibility factor Pls: $2^\theta \rightarrow [0; 1]$, such that:

$$Pls(H) = 1 - Bel(H') = 1 - \Sigma(X \subseteq H')\ cf(X) ;$$

- the ignorance factor Igr : $2^\theta \rightarrow [0; 1]$, such that:

$$Igr(H) = Pls(H) - Bel(H) ;$$

- the certainty range EI : $2^\theta \rightarrow [0; 1] \times [0; 1]$, such that:

$$EI(H) = [Bel(H), Pls(H)] = [\Sigma(X \subseteq H)\ cf(X), 1 - \Sigma(X \subseteq H')\ cf(H')] .$$

By means of certainty factors, we are also able to estimate how much credit can be granted to the evidence which is internally inconsistent. In such a case, it is sufficient to apply the following Dempster's rule of combination:

$$cf_1 \oplus cf_2(Z) = \Sigma(X \cap Y = Z)\ cf_1(X) * cf_2(Y) ,$$

in which X and Y are any input hypotheses, Z is a result hypothesis, $cf_1(X)$, $cf_2(Y)$ and $cf_1 \oplus cf_2(Z)$ mean the certainty factors of, respectively X, Y and Z.

Uncertainty concerns not only hypotheses (also called: facts). It can also be used for rules, which specify dependencies of some facts (conclusions) on other facts (conditions). A rule-based system with uncertainty is a system in which the knowledge base consists of uncertain facts and uncertain rules. In such a system one can keep only inexact reasoning. Systems with uncertainty are very popular among all of the expert systems.

3 Uncertainty and Medicine

An important domain of uncertainty application is medicine. Let us remark that the sick suffer from the same symptoms of a disease, however, not always in the same way. They report on their feelings, again, not always in the same manner. That is why doctors' knowledge about patients and their diseases is uncertain. Uncertainty is also an inherent quality of all medical rules of diagnosis, treatment and prognosis. The diagnosis rule expresses dependencies among symptoms and a hypothetical disease. The treatment rule defines a hypothetical efficiency and hypothetical side effects of a pharmacotherapy for specified symptoms of a disease. The prognosis rule defines a hypothetical health improvement after a one-year pharmacotherapy.

Methods of an uncertain knowledge representation and methods of an inexact reasoning have been implemented in some medical rule based systems, e.g.: MYCIN [6] (designed to help haematologists and general practitioners in diagnosing and treating blood infections) and ONCOCIN/OPAL [3] (acting as an expert in choosing a patient-specific cancer therapy, including pharmacotherapy, chemotherapy, radiotherapy and surgery).

Very rarely such systems are able to provide medical prognoses. Meanwhile, knowing a long-term prognosis is an urgent necessity. Its importance is especially great in case of chronic diseases. The sick would like to know the answers to the questions like these:

- what is a probability of reducing burdensome symptoms of a disease after a one-year treatment by means of some drugs combination?
- what is a possibility of replacing the drug of a high toxicity by the other one of much less toxicity after a one-year treatment?

Let us consider a rule-based system with uncertainty being able to answer these and similar questions. Its knowledge base consists of many constant rules and a few changing facts. The main rules define dependencies between the current level of symptoms intensity and the pharmacotherapy (conditions), and the expected level of symptoms intensity after this pharmacotherapy (conclusions). The sets of symptoms in the rule conditions and conclusions may differ, both in cardinality and in contents. Here we present the example of a prognosis rule:

```
(defrule prognosis-73
     (declare (CF 0.7))
     (symp1-before lev1-3)
   (symp3-before lev3-2)
   (assump1)
   (assump2)
   (pharma med)
 ⇒
   (bind ?symp2-after ((lev2-1  lev2-2)  0.9))
   (bind ?symp1-after ((lev1-2)  0.8)))
```

This rule tells us with a certainty `0.7` that a patient with:

- the level `lev1-3` of the symptom `symp1` and
- the level `lev3-2` of the symptom `symp3`,

under:

- the assumption `assump1` and
- the assumption `assump2`,

after usage:

- the drugs combination `med`

has:

- the level `lev2-1` or `lev2-2` of the symptom `symp2` with a certainty `0.9` and
- the level `lev1-2` of the symptom `symp1` with a certainty `0.8`.

The main facts define a case of the disease. They tell us about a level of the most important symptoms of the disease (and, optionally, about some external conditions of the disease), and a kind of pharmacotherapy recommended. An exemplary set of facts may have a form:

```
(assert (assump1) CF 1.0)
(assert (symp1-before lev1-3) CF 0.8)
(assert (symp2-before lev2-1) CF 1.0)
(assert (symp4-before lev4-3) CF 0.65)
(assert (symp3-before lev3-2) CF 0.9)
(assert (assump2) CF 0.8)
(assert (pharma med) CF 1.0)
```

While reasoning, new facts are added to the above set. They inform about an expected level of symptoms of the disease after one year of treatment.

4 Inexact Reasoning

In a rule-based expert system, the rule is active at the moment if and only if its priority (maybe varying with the progress of reasoning!) is not lower than a threshold factor. Only active rules can be applied while reasoning. A set of active rules, called an agenda, often consists of many different rules. A strategy of choosing the best one at the moment has a significant influence on a final result of reasoning. We can think of the agenda as a set of rules with conclusions being in conflict with one another. Conflicts in the agenda can be resolved in favour of the rules of the highest priorities. These priorities can be granted statically while projecting a knowledge base, or dynamically on the grounds of certainty factors. Other criteria of the choice may be: the specificity of rules (the rules that have more conditions are in preference to those with fewer conditions) or the recency of conditions (the rules which apply to the most recently deduced facts are preferred to those which apply to the older facts).

One of the most known algorithms of inexact reasoning is the Dempster's rule of combination. Let us present this algorithm in reference to medical prognosticating.

4.1 Dempster's Rule of Combination

Doctors are entitled to demand that a medical knowledge base expresses large views upon a disease, methods of its treatment and the chances that the treatment will succeed. These views usually differ from one another. In consequence, each symptom of the disease may be found in many rules: in various contexts as well as with various certainty factors. The resultant certainty factors of conclusions are calculated in the process of a cyclic approximation. An algorithm of this approximation has an effect both on values of the resultant certainty factors and on the time necessary for calculations.

In a case of medical prognosticating, the set of active rules does not change during the whole process of reasoning. In consequence, the set of ultimate conclusions is constant.

The Dempster's rule of combination takes into account all the conclusions which concern the same domain (in the case - the same symptom of a disease). Let us assume that A, B, C and D mean the facts informing about the four different levels of the symptom `symp1` of a disease after a one-year pharmacotherapy, respectively: `(symp1-after lev1-1)`, `(symp1-after lev1-2)`, `(symp1-after lev1-3)` and `(symp1-after lev1-4)`. {A, B, C, D} is the set of mutually exclusive and exhaustive hypotheses regarding the level of `symp1` after a one-year pharmacotherapy. We call it the environment and denote by means of Θ_1.

Let us assume that the first rule in the agenda is `prognosis-73` and the second one is `prognosis-17` as follows:

```
(defrule prognosis-17
  (declare (CF 0.8))
  (symp1–before lev1-3)
  (symp4–before lev4-3)
  (pharma med)
 ⇒
  (bind ?symp1-after ((lev1-2  lev1-3)  0.9))
  (bind ?symp1-after ((lev1-1)  0.1)))
```

In such a case, the calculations of the `symp1` value will be performed according to the table 1. In consequence, after performing the normalization we obtain:

$cf_{73}\oplus cf_{17}\ (A) = (0.04/96)*100 \approx 0.04$
$cf_{73}\oplus cf_{17}\ (B) = (0.51/96)*100 \approx 0.53$
$cf_{73}\oplus cf_{17}\ (B,C) = (0.32/96)*100 \approx 0.33$
$cf_{73}\oplus cf_{17}\ (\Theta_1) = (0.09/96)*100 \approx 0.10$
$Bel_{73,17}(A) = 0.04$
$Pls_{73,17}\ (A) = 1 - (0.53 + 0.33) = 0.14$
$Igr_{73,17}(A) = 0.14 - 0.04 = 0.10$
$Bel_{73,17}(B) = 0.53$
$Pls_{73,17}\ (B) = 0.96$
$Igr_{73,17}(B) = 0.96 - 0.53 = 0.43$
$Bel_{73,17}(C) = 0$
$Pls_{73,17}(C) = 1 - (0.53 + 0.04) = 0.43$
$Igr_{73,17}(C) = 0.43 - 0 = 0.43$

Table 1. Calculations of the `symp1` value

	$cf_{73}(\{B\}) = 0.56$	**$cf_{73}(\Theta_1) = 0.44$**
$cf_{17}(\{A\}) = 0.08$	**{ } 0.04**	**{A} 0.04**
$cf_{17}(\{B,C\}) = 0.72$	**{B} 0.40**	**{B,C} 0.32**
$cf_{17}(\Theta_1) = 0.20$	**{B} 0.11**	**Θ_1 0,09**

4.2 Rule of Convergence

Let us assume that the knowledge base consists of *n* rules. For each resultant level of the symptom of a disease the Dempster's rule of combination has the same linear complexity O(n). Let us remark that the number *n* is very large (it may be several hundred or even several thousand) and the number of levels is not lower than 16 (there are usually four levels in each of the four main symptoms of the disease). Considering that, it is impossible to obtain a real time implementation of this rule. Furthermore, such a prognosis could be done for several chosen drug combinations!

How to speed up reasoning in a system with uncertainty? To cope with this problem, the special rule of convergence is proposed. Wherein lies the essence of this rule?

We can assume that, as in the case of the Dempster□s rule of combination, the inference engine does forward chaining. Additionally, we can assume that the conflicts in the agenda are resolved in favour of the rules of the highest priorities. Priority of a rule is calculated dynamically from the following equation:

$$pr(RULE) = (\min(1 \le i \le n)\ cf(C_i)) * cf(RULE)\ ,$$

where RULE is a rule of a rule-based system with uncertainty, pr (RULE) means the priority of RULE, cf(RULE) means the certainty factor of RULE, $cf(C_i)$ means the certainty factor of the i-th RULE condition, min is the function of minimum.

At the beginning the rule of convergence calculates the values $\mathrm{Bel}_h(X)$ and $\mathrm{Pls}_h(X)$ for the `prognosis-h` rule of the highest priority and each resultant level X of the main symptom of a disease. The next calculation must be done for the rule of the succeeding priority, on the assumption of the restricted certainty range: $[\mathrm{Bel}_h(X) ; \mathrm{Pls}_h(X)]$ (instead of the previous range [0 ; 1]).

For the two mentioned above rules: `prognosis-73` and `prognosis-17`, the following values will be obtained:

– in the first step of algorithm:

$$\mathrm{Bel}_{73}(B) = 0.56 \qquad \mathrm{Pls}_{73}(B) = 1 \qquad \mathrm{Igr}_{73}(B) = 0.44 ,$$

– and in the second step

$$\mathrm{Bel}_{73,17}(B) = 0.56 \qquad \mathrm{Pls}_{73,17}(B) = 0.96 \qquad \mathrm{Igr}_{73,17}(B) = 0.40 .$$

The above procedure must be continued until the agenda is empty or the relation $\mathrm{Igr}(X) \leq \varepsilon$ is satisfied, where ε means some permissible degree of ignorance. Let us remark that a sequence of ignorance values obtained in such a way converges at zero. In consequence, these calculations are usually much shorter than the former ones.

The proposed technique depends strongly on the method of agenda conflict resolution. To confirm this statement, let us assume the idealized situation in which the application of any rule in the process of reasoning reduces the ignorance factor for a conclusion by 10%. Then, with the required accuracy of: 0.1, 0.01, 0.001 and 0.0001, the conclusion certainty factor can be determined after applying: 22, 44, 66 and 88 rules, respectively.

If we assume that the ignorance factor reduces with the constant speed of 20%, the number of indispensable rules for the same accuracies will equal: 11, 22, 33 and 42, respectively.

Now, if we assume that the number of active rules may reach the level of several hundred, it will result from the above considerations that, applying the method of convergence, only a small percentage (about 10% - 30%) out of all active rules will be used. The choice of rules for this set has an obvious influence on determining the certainty factors of conclusions. Therefore, it is important to consider this choice very carefully, and not to make it at random. Such a choice □as it was mentioned above □ may be easily guaranteed thanks to the application of an algorithm which recognizes the rule priority criterion as the principal one. Then, we can be sure that the most important rules (the most authoritative) will have a decisive character, whereas the other ones, less essential, will be actually placed beyond the agenda (they will not take part in the "shortened" reasoning process, and they will not have any influence on the certainty factors of conclusions).

In fact, we can expect that the rules stored in the system knowledge base will draw similar, and not contradictory, conclusions from the similar conditions. To prevent the occurrence of contradictions in the knowledge base, there should exist an efficient truth maintenance system.

However, the rules presented above, `prognosis-73` and `prognosis-17`, could certainly coexist in the knowledge base. It appears that a slight change of their priorities (e.g. increasing the priority of the `prognosis-17` from `0.8` to `0.86`) will exhort a different course of the reasoning process and, in consequence, will give a new certainty factor of the conclusion. In such case, if we assume that the rules:

`prognosis-73` and `prognosis-17` will be activated at the beginning of the reasoning process, but in the reverse order to the previous one, we will obtain the following values of the factors:

- in the first step of algorithm:

  $Bel_{17}(B) = 0$ $Pls_{17}(B) = 0.92$ $Igr_{17}(B) = 0.92$,

- and in the second step:

  $Bel_{17,73}(B) = 0.52$ $Pls_{17,73}(B) = 0.92$ $Igr_{17,73}(B) = 0.40$.

We shall notice that the rules activation order, within the chosen set of rules, does not influence the final value of the conclusion ignorance factor. This fact can be used while implementing the convergence algorithm.

Obviously, in the Dempster's rule of combination, the method of conflict resolution has no influence on the final certainty factor value at all.

4.3 Exemplary Prognosticating in Asthma Disease

Nowadays, one of the most widespread diseases is bronchial asthma. It is a chronic respiratory disease, affecting people all over the world. You can fall ill with asthma if your immune system is bad. Such a system recognizes environmental factors (grass-pollen, house-dust, noxious gases, etc.), food components or microbes as enemies, dangerous for human health and life. When an asthmatic patient comes into contact with them, his immune system shows hyper-responsiveness: it raises an allergic chain response, with a violent bronchus spasm at the last. According to the progress of the disease, we distinguish the following groups of bronchial asthma:

- occasional asthma (I),
- chronic asthma of mild progress (II),
- chronic asthma of medium progress (III),
- chronic asthma of severe progress (IV).

The treatment of asthma disease is a long-term, complex task, to which both doctors and patients themselves have to contribute. For instance, a great effect on the reduction of bronchial hyperreactivity has a prolonged allergen avoidance. Also, the asthmatic patients are often required to do some physical exercises or even some sports to increase their lung volume [4]. However, the most important component of treatment is still pharmacotherapy [1]. Approximately, the basic scheme of this pharmacotherapy is very much the same for all the bronchial asthma groups.

The four main symptoms of bronchial asthma are: a night-time cough, an exercise-induced lung inefficiency, over wheezing, dragging on (for above 10 days) airway infections. In order to decrease the levels of symptoms intensity, an asthmatic patient is required to take the combination of 3-4 drugs, which have different effects on his or her body. Most often this combination consists of histamine antagonist, leukotriene receptor antagonist and maintenance medication (long-acting theophylline or short-acting beta2-agonist). Additionally, in a case of an airway infection, an anti-inflammatory drug could be given (an oral steroid or an inhaled corticosteroid).

While medical knowledge is developed enough to diagnose bronchial asthma (i.e. estimate the levels of the four main symptoms of the disease and – optionally – detect its additional symptoms) and to choose a patient-specific therapy (in particular pharmacotherapy), we can hardly foresee the detailed effects of this treatment. Drugs – separately or jointly – decrease the levels of the symptoms of the disease or even suppress some of them. Their final influence, however, cannot be certainly known: it can only be estimated with some degree of certainty. For instance, we can say with a certainty factor of `0.7` that a patient initially classified to group III of bronchial asthma, with a night-time cough of frequency of several times a week □ after prolonged (over one year) taking of Singulair □has a night-time cough of once a week frequency at 80 per cent and a night-time cough of the former frequency at 20 per cent:

```
(defrule prognosis-100
  (declare (CF 0.7))
  (cough-before levc-3)
  (group 3)
  (pharma comb(Singulair, xx))
⇒
  (bind ?cough-after ((levc-2)  0.8))
  (bind ?cough-after ((levc-3)  0.2))
)
```

Thus, considering a night-time cough, Singulair turns out to be ineffective at 20 per cent. It does not mean that at 20 per cent Singulair has no influence on frequency of other symptoms of a disease.

5 Conclusions

With a reference to the above prognosticating rules, if an expert system can reason efficiently, the doctor will draw a comparison of various pharmacotherapies at once. In consequence, he will be able to choose the best possible pharmacotherapy.

A medical system capable of efficient reasoning can prepare both a short-term (one-year) and a long-term (several-year) prognosis (the latter one through a cyclic execution of the algorithm under consideration). Keeping in mind that the longer prognosis the lesser its reliability, we can still answer many nontrivial questions, e.g.:

- what is a probability that after a one-year (two-year) pharmacotherapy by means of some drugs combination the cough frequency will considerably decrease? the lung efficiency will somewhat improve?
- how certain it is that after a one-year treatment it will be possible to replace a strong steroidal drug by another one of much less toxicity?
- is it likely that a probability of making a progress towards a cure of asthma (i.e. a probability of being classified to a lower group of bronchial asthma) will reach the level of 50 per cent after a three-year pharmacotherapy?

The algorithms of making prognoses about the bronchial asthma progress are being implemented in the RiAD system (Relief in Asthma Desease) [2,7]. So far, the system has been equipped with modules for making diagnoses and choosing a patient-specific pharmacotherapy. We are going to implement the rule of convergence in RiAD. We expect it to give us a possibility of a real-time prognosticating. Up till now the language used for RiAD implementation has been FuzzyCLIPS [5].

The rule of convergence can be applied in all the expert systems in which speedy, even if approximate answer is needed. We mean here the systems of diagnostics and classification (e.g. automobile engine diagnostics, laboratory data interpretation, plants identification or recognition of geological formations) and the systems of simulation (e.g. factory weak-line analysis, electrical circuit simulation or weather forecasting).

References

1. Deutsch T., Cramp D., Carson E., Decision, Computers and Medicine: The Informatics of Pharmacotherapy (Elsevier Science, Amsterdam, 2001)
2. Jankowska B., Zastosowanie metod inżynierii wiedzy w prognozowaniu efekt□w leczenia astmy oskrzelowej, in: Proc. IV Krajowa Konferencja Naukowa SzI-15□2000 (Siedlce - Warszawa, Poland, September 2000) 341-348
3. Musen M. et al., OPAL: Use a Domain Model to Drive an Interactive Knowledge-Editing Tool, in: International Journal of Man-Machine Studies 26 (1987) 105-121
4. Ogl□no światowa strategia leczenia astmy oskrzelowej i jej prewencji. Raport NHBI/WHO (National Institutes of Health, Publication Number 95-3659, January 1995)
5. Orchard R.A., FuzzyCLIPS Version 6.04A. User's Guide (National Research Council, Canada, October 1998)
6. E. Shortliffe, Computer-Based Medical Consultations: MYCIN (American Elsevier, 1976).
7. S. Wr□bel, Zastosowanie metod in żynierii wiedzy w procesach prognozowania (Master's Thesis, Poznań University of Technology, Poznań, July 2000)

Efficient BDD Encodings for Partial Order Constraints with Application to Expert Systems in Software Verification

Masahito Kurihara[1] and Hisashi Kondo[2]

[1] Hokkaido University, Sapporo, 060-8628, Japan
kurihara@main.eng.hokudai.ac.jp
[2] Ibaraki University, Hitachi, 316-8511, Japan
H.Kondo@dse.ibaraki.ac.jp

Abstract. We introduce a class of computational problems called the partial order constraint satisfaction problems (POCSPs) and present three methods for encoding them as binary decision diagrams (BDDs). The first method, which simply augments domain constraints with the transitivity and asymmetry for partial orders, is improved by the second method, which introduces the notion of domain variables to reduce the number of Boolean variables. The third method turns out to be most useful for monotonic domain constraints, because it requires no explicit encoding for the transitivity. We show how those methods are successfully applied to expert systems in a software verification domain.

1 Introduction

Binary decision diagrams (BDDs) [1, 3, 18] are graphical, compact representation of Boolean functions, successfully applied in various fields of industrial applications of AI and computer science. For example, they are applied in the field of VLSI design expert systems [10], truth maintenance systems [9], and the reliability analysis of huge and complex plant systems [4]. They also interest the community of automated theorem proving [12].

However, it should be stressed that although this technology is quite general and may seem to be easily applied in various fields, it is known that it essentially contains some computationally hard problems to be solved, depending on the application domains. One of such problems is to determine an appropriate variable order for constructing BDDs. Without good variable orders, the size of the BDDs could exponentially grow too large. For the theoretical computational complexity results, consult [2,14,17] and for some of the heuristic variable ordering methods, see [7,10,11].

Another problem is how to encode the problems at hand as Boolean functions, thus as BDDs. Recently, we have developed the framework of applying BDD technology in the field of development of rule-based programs, but it does not necessarily imply a success in that field: we need heuristics. In this paper, we present some heuristic approaches to this problem and evaluate them through comprehensive experiments on sample rule-based programs taken from practical domains such as hardware diagnosis, software specification, and mathematics. The results show the big difference among those approaches and provide us useful information for optimizing the overall sys-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 827-837, 2004.

tems. Such information has been integrated into the expert system for verifying the termination of rule-based programs.

In Section 2, the key ideas are described in an abstract setting of partial order constraint satisfaction problems (POCSPs). In Section 3, we present three methods for encoding POCSPs as BDDs. In Section 4, the three methods are applied to the mechanical verification of the termination of rule-based programs. The experimental results show the big difference in efficiency among the methods.

2 Partial Order Constraint Satisfaction Problems

A *binary decision diagram* (BDD)[1,3,18], notation BDD(F), is a graphical, compact representation of a Boolean function F. It is like a tree except that several vertices may be structurally shared in the graph. Given an assignment of Boolean values (0 or 1) to each variable of F, we can determine the value of the function by following the path from the root to a terminal (0 or 1), branching at each x_i-labeled node to either 0- or 1-labelled edge depending on the assigned value for x_i.

Usually, we fix a linear order (called a *variable order*) on the set of variables, and for every path from the root to a terminal, we let these variables appear in this ascending order. When the variable order is fixed, the BDD of the given function is uniquely determined. As a result, the satisfiability of a Boolean function can be determined simply by checking if its BDD representation is different from BDD(0).

A *constraint satisfaction problem* (CSP) $P=(X, D, C)$ consists of a finite set of n *variables* $X=\{x_1,\ldots,x_n\}$, a set of finite *domains* $D=\{D_1,\ldots,D_n\}$, and a set of *constraints* C among variables. Each variable x_i must take a value from the set D_i. Every constraint is associated with some variables and specifies the *consistent* (allowed) combinations of values for those variables. A solution of P is an assignment of values to all n variables such that it is consistent with all the constraints.

In some applications, there is often a need to mechanically determine a partial order that must satisfy some constraints. This motivates the following definition.

Definition 1. Let S be a finite set of the elements called symbols. Then a *partial order constraint satisfaction problem* (POCSP) over S is a CSP $P_S = (X, D, C)$, where $X=\{X_{fg} \mid f, g \in S, f \neq g\}$ is a set of Boolean variables with the common domain $D=\{0,1\}$, and C consists of the three kinds of constraints as follows.

- the *transitivity* constraints: if $X_{fg}=1$ and $X_{gh}=1$ then $X_{fh}=1$ for all combinations (f, g, h) of three mutually-distinct symbols;
- the *asymmetry* constraints: either $X_{fg}=0$ or $X_{gf}=0$ for all pairs (f, g) of distinct symbols; and
- the *domain constraints* C_1 □ C_m.

The POCSP is seen as a problem of determining a partial order $>$ on the set of symbols S under the domain constraints. It is straightforward to encode such a problem as a POCSP. We represent the relationship $f > g$ by an assignment of a Boolean value to X_{fg}. We let X_{fg}=1 if $f > g$, and X_{fg}=0 otherwise. The relation $>$ is represented by an assignment of truth values to all the variables in X. Such an assignment *represents* $>$. If $>$ is a strict partial order, the assignment must satisfy the transitivity and asymmetry. Together with the domain constraints, we get a POCSP.

3 BDD Encodings for POCSPs

We present three methods for encoding POCSPs as Boolean functions, thus as BDDs. The first one is a basic one derived directly from the definition of POCSPs. The second encoding improves it by reducing the number of Boolean variables. When the domain constraints are *monotonic*, the last encoding can safely remove the transitivity part at the (expectedly small) cost of losing the minimality of the asymmetry.

3.1 Basic Encoding

The transitivity and asymmetry of Definition 1 are represented by the following Boolean functions, respectively.

$$T(X) = \prod_{f,g,h\in S, f\neq g\neq h\neq f} (\overline{X}_{fg} + \overline{X}_{gh} + X_{fh}) \tag{1}$$

$$A(X) = \prod_{f,g\in S, f\neq g} (\overline{X}_{fg} + \overline{X}_{gf}) \tag{2}$$

We assume that each domain constraint C_i ($i=1,\dots,m$) is also encoded as a Boolean function $P_i(X)$ that takes the value 1 if and only if the assignment to X satisfies C_i. The overall domain constraint is represented by the following function:

$$P(X) = \prod_{i=1}^{m} P_i(X) . \tag{3}$$

Now we combine the three functions to yield the basic encoding:

$$B(X) = (T \cdot A \cdot P)(X) \tag{4}$$

where the right-hand side is a shorthand notation for $T(X) \cdot A(X) \cdot P(X)$.

Theorem 1. *The POCSP has a solution if and only if $B(X)$ is satisfiable.*

Example 1. Consider a set $S = \{f, g, h\}$. Then we have

$$T(X) = (\overline{X}_{fg} + \overline{X}_{gh} + X_{fh}) \cdot (\overline{X}_{fh} + \overline{X}_{hg} + X_{fg}) \cdot (\overline{X}_{gf} + \overline{X}_{fh} + X_{gh})$$
$$\cdot (\overline{X}_{gh} + \overline{X}_{hf} + X_{gf}) \cdot (\overline{X}_{hf} + \overline{X}_{fg} + X_{hg}) \cdot (\overline{X}_{hg} + \overline{X}_{gf} + X_{hf}),$$
$$A(X) = (\overline{X}_{fg} + \overline{X}_{gf}) \cdot (\overline{X}_{gh} + \overline{X}_{hg}) \cdot (\overline{X}_{hf} + \overline{X}_{fh}).$$

The domain constraints may be given arbitrarily, but for now, assuming that the problem requires that either $f > g$ or $h > g$, we have $P(X) = X_{fg} + X_{hg}$.

3.2 Encoding with Domain Variables

If the set S contains n symbols, $T(X)$ consists of $O(n^3)$ clauses and tends to be the biggest part of $B(X)$. In most cases, however, the set of Boolean variables actually occurring in the domain constraint part $P(X)$ tends to be only a small subset of X. We call them *domain variables*. This motivates the following definitions.

Consider a POCSP over S, and let X_D be the set of domain variables occurring in $P(X)$. Consider a graph $G = (V, E)$ with $V = S$ and $E = \{(f, g) \mid X_{fg} \in X_D\}$. Each vertex $f \in V$ is a symbol of S and each directed edge $(f, g) \in E$ corresponds to a domain variable X_{fg}. We call G the *domain graph*.

Definition 2 (Domain Transitivity and Asymmetry). Suppose that the domain graph $G = (V, E)$ contains an edge $(f, g) \in E$ such that there is a path $f = v_0\ \ v_1 \ldots\ v_n = g$ from f to g with length $n \geq 2$. Then the domain transitivity condition derived from this path is $\overline{X}_{fv_1} + \ldots + \overline{X}_{v_{n-1}g} + X_{fg}$. The product of such conditions for all edges $(f, g) \in E$ and for all paths from f to g with length $n \geq 2$ is called the *domain transitivity* for X_D.

Suppose that G contains a cycle $v_0\ \ v_1 \ldots\ v_n = v_0$ of length $n \geq 2$. Then the domain asymmetry condition derived from this cycle is $\overline{X}_{v_0 v_1} + \ldots + \overline{X}_{v_{n-1} v_0}$. The product of such functions for all cycles contained in G is called the *domain asymmetry* for X_D.

A path is *minimal* if it is elementary (i.e., does not contain any vertex twice) and contains no proper subpath that can be replaced by a bypass edge (or chord) in E. A cycle is *minimal* if it is elementary (i.e., does not contain any vertex twice, except that the start and the end vertices are the same) and contains no subpath that can be replaced by a bypass edge (or chord) in E. The following lemma ensures that we need to enumerate only minimal conditions.

Lemma 1. *All the domain transitivity and asymmetry conditions are satisfied if all the minimal conditions are satisfied.*

The domain transitivity and asymmetry conditions derived only from minimal paths and cycles are denoted by $T^D(X_D)$ and $A^D(X_D)$, respectively. In practice, the graph G is often sparse, and enumeration of minimal paths and cycles can be performed in practical time and space. Now we introduce the second encoding:

$$B^D(X_D) = (T^D \cdot A^D \cdot P)(X_D) \tag{5}$$

where the notation $P(X_D)$ for $P(X)$ emphasizes that all the variables occurring in $P(X)$ are members of X_D by definition.

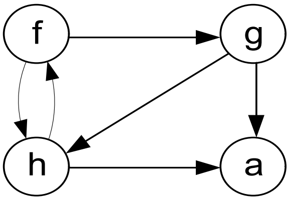


Fig. 1. The domain graph for Example 2

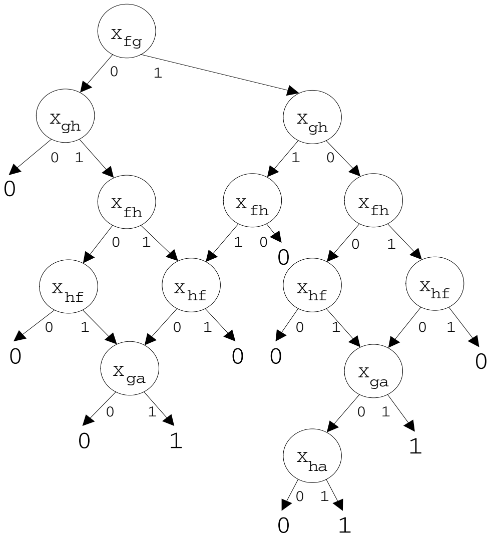


Fig. 2. The BDD for Example 2

Example 2. Let $S = \{f, g, h, a\}$ and suppose that the domain constraints are encoded as $P(X) = (X_{fg} + X_{gh})(X_{fh} + X_{hf})(X_{ga} + X_{ha})$. Then X_D consists of 6 variables, and the domain graph is given in Fig. 1. The domain transitivity and asymmetry are given as follows.

$$T^D(X_D) = (\overline{X}_{fg} + \overline{X}_{gh} + X_{fh})(\overline{X}_{gh} + \overline{X}_{ha} + X_{ga})$$
$$A^D(X_D) = \overline{X}_{fh} + \overline{X}_{hf}.$$

Note that the cycle $f\ g\ h\ f$ is non-minimal, because there is a bypath edge (f, h). The BDD representation for $B^D(X_D)$ is given in Fig. 2.

Theorem 2. *$B^D(X_D)$ is satisfiable if and only if $B(X)$ is satisfiable.*

This theorem implies that we can search for the solution of the POCSP by considering $B^D(X_D)$ instead of $B(X)$. Since $B^D(X_D)$ contains only domain Boolean variables, it encodes the POCSP with a smaller number of Boolean variables. This is important when representing the function by the BDD, because its size is likely to grow rapidly (exponentially) as the number of variables increases.

3.3 Encoding without Transitivity for Monotonic Domain Constraints

Since the size of transitivity conditions is often so big even for domain transitivity, it can cause an efficiency problem. We can be more efficient if the domain constraint $P(X_D)$ is monotonic. A Boolean function $F(X)$ is *monotonic* if $\alpha \le \alpha'$ implies $F(\alpha) \le F(\alpha')$, for all assignments α and α'. In particular, we can verify that every Boolean function constructed from the sum and the product (thus containing no negations) is monotonic. In such cases, we can remove the transitivity conditions at the (expectedly small) cost of additional complexity in domain asymmetry.

Let $A^*(X_D)$ be the domain asymmetry conditions for X_D obtained by considering all minimal cycles plus all non-minimal elementary cycles (that have bypass edges). We introduce the following function as our third encoding:

$$B^*(X_D) = (A^* \cdot P)(X_D)\,. \tag{6}$$

Example 3. Recall Example 2. Since the domain constraint $P(X_D)$ is monotonic, we can employ the third encoding by considering all cycles to get

$$A^*(X_D) = (\overline{X}_{fh} + \overline{X}_{hf})(\overline{X}_{fg} + \overline{X}_{gh} + \overline{X}_{hf}).$$

The BDD representation for $B^*(X_D)$ is given in Fig. 3. The BDD has 9 nodes, compared with 16 nodes of Fig. 2.

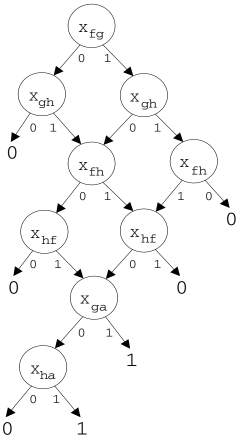


Fig. 3. BDD for Example 3

When the domain graph contains a relatively small number of non-minimal cycles, the size of the representation for $B^*(X_D)$ is often significantly smaller than $B(X)$ and even than $B^D(X_D)$.

$B^*(X_D)$ contains no transitivity conditions. Nevertheless, we can prove the following theorem that ensures that when the domain constraint $P(X_D)$ is monotonic, we can use this function instead of $B(X)$ and $B^D(X_D)$.

Theorem 3. *Suppose $P(X_D)$ is monotonic. Then $B^*(X_D)$ is satisfiable if and only if $B^D(X_D)$ is satisfiable.*

4 Application to Software Verification

4.1 Termination of Rule-Based Programs

Verification of correctness of computer programs is one of the most challenging applications of AI. We present an application of POCSP to verification of the termination property of programs written in a language called *term rewriting systems* [5,13]. A *term* refers to an expression constructed from *function symbols* and *term variables*. A program R is a set of rewrite rules consisting of a pair of terms $l \to r$ specifying that instances of l in a term can be rewritten to the corresponding instance of r. Given a term input from the users, the program uses a pattern matching algorithm to rewrite it repeatedly until one gets an answer term, which is a term that cannot be rewritten any more.

A program is *terminating* if there are no infinite rewrite sequences. Termination is an undecidable property in general, but some sufficient conditions for its verification have been studied. In this paper, we focus on a popular class of such conditions based on precedence. A *precedence,* denoted by $\succ$, is a partial order on the set of function symbols. It induces the partial order $\succ_{lpo}$ (called the lexicographic path order, or lpo) on the set of terms. The definition of the lpo is omitted, but all one has to know in this paper is that once you determine the precedence, the lpo is uniquely determined.

Theorem 4. *If there exists a precedence $\succ$ such that $l \succ_{lpo} r$ for all rewrite rules $l \to r$ of R, then R is terminating.*

With two terms s and t fixed, the truth of $s \succ_{lpo} t$ is uniquely determined by $\succ$. Let $P_{s,t}(X)$ be the Boolean function which takes 1 if $s \succ_{lpo} t$ and takes 0 otherwise, provided the assignment to X represents $\succ$. Based on the definition of $\succ_{lpo}$, this function is defined recursively with respect to term structure as follows.

Definition 3. Let s and t be two terms. When s and t are not term variables, we write $s = f(s_1,\dots,s_m)$ and $t = g(t_1,\dots,t_n)$, respectively. Then:

$$P_{s,t}(X) = \begin{cases} 0 \quad \mathit{if}\ s = t\ \mathit{or}\ s \in V\ \mathit{or}\ (t \in V, t \notin Var(s)) \\ 1 \quad \mathit{if}\ (s \notin V, t \in Var(s))\ \mathit{or}\ \exists i.s_i = t \\ \sum_{i=1}^{m} P_{s_i,t}(X) + X_{fg} \cdot \prod_{j=1}^{n} P_{s,t_j}(X) \\ \quad \mathit{if}\ s \notin V, t \notin V, f \neq g, \forall i.s_i \neq t \\ \sum_{i=1}^{m} P_{s_i,t}(X) + P_{s_k,t_k}(X) \cdot \prod_{j=k+1}^{n} P_{s,t_j}(X) \\ \quad \mathit{if}\ s \notin V, t \notin V, f = g, s \neq t, \forall i.s_i \neq t \end{cases} \tag{7}$$

where k is the minimal value of i ($1 \le i \le n$) satisfying $s_i \neq t_i$; V is the set of term variables and $Var(V)$ is the set of term variables contained in s.

Example 4. Let $s = f(h(x))$ and $t = g(x)$. Then we have

$$\begin{aligned} P_{s,t}(X) &= P_{h(x),g(x)}(X) + X_{fg} \cdot P_{f(h(x)),x}(X) \\ &= [P_{x,g(x)}(X) + X_{hg} \cdot P_{h(x),x}(X)] + X_{fg} = X_{hg} + X_{fg}. \end{aligned}$$

Given a program $R = \{l_i \to r_i \mid 1 \le i \le m\}$, the following Boolean function represents the domain constraints that $l_i \succ_{lpo} r_i$ for all rules in R:

$$P(X) = \prod_{i=1}^{m} P_{l_i,r_i}(X).$$

Let us define the following encoding of POCSP:

$$B(X) = (T \cdot A \cdot P)(X).$$

Theorem 5. *If $B(X)$ is satisfiable, then R is terminating.*

Note that $P(X)$ is monotonic. Therefore, as discussed in the previous section, the satisfiability of $B(X)$ is equivalent to the satisfiability of $B^*(X_D) = (A^* \cdot P)(X_D)$.

4.2 Experiments and Comparison

We restrict ourselves to the following three schemes for determining precedences. The *random order* (RAND) just orders the variables randomly, according to the uniform distribution. The *generation order* (GO) just orders variables as generated by our BDD generation program. More precisely, it is the order in which we encounter the Boolean variables while expanding $P_{l,r}(X)$ for each input rule in a depth-first, left-to-right manner according to the definition. The *reversed generation order* (RGO) is just the orders obtained by reversing GO.

We have performed comprehensive experiments on typical sample programs taken from several application domains, including hardware diagnosis, software specification, and mathematics. In this section, we present some results to show how the efficiency is affected by the choice of variable orders.

Effects by Variable Orders

We start by fixing our attention to the experimental results for $B^D(X_D)$. We have selected two particular problems for presenting the results in detail. One is taken from the field of the model-based hardware diagnosis. We refer to this problem as CIRCUIT [8]. The rules have been introduced by directing, from left to right, the 24 equations that specify the behavior of the full adder circuit. The number of domain Boolean variables for this problem is 36.

The other problem referred to as SOLITAIRE consists of 28 rules taken from the field of the algebraic specification of software systems [15]. The number of domain Boolean variables for this problem is 46.

The results, summarized in Tables 1 and 2, show that the GO and RGO are more efficient than the RAND in both time and space. We recommend the use of RGO, because in most cases it shows the best performance in time and space.

Table 1. Results for circuit

Var order	RAND	GO	RGO
BDD size	39	33	33
CPU time	9533	797	70

Table 2. Results for solitaire

Var order	RAND	GO	RGO
BDD size	1459	398	400
CPU time	2479	452	74

Table 3 shows the results for five other problems taken from the domain of mathematics. Those are all the problems given in [16] that contain at least five function symbols and are terminating. The variable order is fixed to RGO. The table includes two new entries for the number of rules and the number of domain variables. We can say that all the problems have been solved efficiently.

Table 3. Results for mathematics

No.	12	26	27	29	31
Rules	9	10	11	7	7
Dom vars	14	18	17	18	11
BDD size	16	15	37	34	15
CPU time	1.2	.94	1.4	.35	.10

Table 4. Results for transitivity removal

No.	Circuit	Solitaire
Rules	24	28
Dom vars	36	46
BDD size	35	93
CPU time	6.95	2.62

Effects by Domain Conditions

We briefly describe how the domain transitivity and asymmetry on X_D result in more efficiency than the ordinary transitivity and asymmetry on X. Indeed, the use of $B^D(X_D)$ is far better than $B(X)$. This is justified by our experiments in which the use of $B(X)$ did not yield a solution of the CIRCUIT problem within 48 hours. Actually, we found that this run required a temporary BDD whose maximum size is 35 times greater than the BDD required for $B^D(X_D)$.

Effects by Transitivity Removal
Table 4 shows the results of the experiments in which the previous two sample problems were solved by using $B^*(X_D)$ in place of $B^D(X_D)$. We only show the entries for the RGO order. Comparing the CPU time in this table with Tables 1 and 2, we can see that the use of $B^*(X_D)$ is far more efficient than $B^D(X_D)$.

5 Conclusion

We have presented three methods for encoding POCSPs as BDDs, and applied and evaluated them in the field of software verification. Let us compare our work with two other related works. One is a most simple way based on backtracking [6]. It is well-known that the simple backtracking suffers from inefficiency caused by futile backtracking, rediscovering contradictions, and rediscovering inferences. Actually, the CPU time for computing all solutions for the CIRCUIT and SOLITAIRE problems by the backtracking method was 108 and 2700 seconds, respectively, compared with 70 and 74 seconds by our method. The other work [8] uses a reason maintenance system to avoid the drawbacks caused by the simple backtracking. It is reported that this method was successful in getting a single solution efficiently. In contrast, our method is effective even if all solutions are sought or if one wants to check that there are no solutions.

Acknowledgement. This work was partly supported by JSPS Grant-in-Aid for Scientific Research (B)(15300188) and MPHPT Strategic Information and Communications R&D Promotion Programme.

References

1. Akers, S. B.: Binary decision diagrams, IEEE Trans. Comput., Vol. C-27, 6, 509□516 (1978)
2. Bollig, B. and Wegener, I., Improving the variable ordering of OBDDs is NP-complete, IEEE Trans. on Computers 45, 993□1002 (1996)
3. Bryant, R. E.: Graph-based algorithm for boolean function manipulation, IEEE Trans. Comput., Vol. C-35, 5, 677□691 (1986)
4. Coudert, O. and Madre, J. C.: Towards an interactive fault tree analyser, Proc. IASTED Int. Conf. on Reliability, Quality Control and Risk Assessment (1992)
5. Dershowitz, N. and Jouannaud, J.-P.: Rewrite Systems, in J. van Leeuwen (ed.), Handbook of Theoretical Computer Science, vol. B, North-Holland, 243□320 (1990)
6. Detlefs, D. and Forgaad, R.: A procedure for automatically proving the termination of a set of rewrite rules, Proc. of 1st Conf. on Rewriting Techniques and Applications, Springer, LNCS 202, 255□270 (1985)

7. Fujita, M., Fujisawa, H., and Matsunaga, Y., Variable ordering algorithms for ordered binary decision diagrams and their evaluation, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 12, 6□12 (1993)
8. Kurihara, M., Kondo, H., et al.: Using ATMS to efficiently verify the termination of rewrite rule programs, Intern. Journal of Software Engineering and Knowledge Engineering, Vol.2, 4, 547□565 (1992)
9. Madre, J. C. and Coudert, O.: A logically complete reasoning maintenance system based on a logical constraint solver, Proc. Intern. Joint Conf. on Artificial Intelligence, 294□299 (1991)
10. Malik, S., et al.: Logic verification using binary decision diagrams in a logic synthesis environment, Proc. IEEE Int. Conf. on Computer-Aided Design, 6□9 (1988)
11. Minato, S., Ishiura, N., and Yajima, S., Shared binary decision diagram with attributed edges for efficient Boolean function manipulation, 27th DAC, 52□57 (1990)
12. Moore, J. S.: Introduction to the OBDD algorithm for the ATP community, Journal of Automated Reasoning, Vol.12, 33□45 (1994)
13. Plaisted, D. A.: Equational reasoning and term rewriting systems, in Gabbay, D. M. (ed.), Handbook of Logic in Artificial Intelligence and Logic Programming, 274□364 (1993)
14. Sieling, D., On the existence of polynomial time approximation schemes for OBDD minimization, STACS□98, LNCS 1373, 205□215 (1998)
15. Steinbach, J.: Termination of rewriting : Ph.D thesis, Univ. Kaiserslautern, Germany (1994)
16. Steinbach, J. and K□hler, U.: Check your ordering : termination proofs and open problems, SEKI report SR-90-25(SFB), Univ. Kaiserslautern, Germany (1990)
17. Tani, S., Hamaguchi, K., and Yajima, S., The complexity of the optimal variable ordering problem of a shared binary decision diagram, ISAAC□93, LNCS 762, 389□398 (1993)
18. Wegener, I.: Branching Programs and Binary Decision Diagrams: Theory and Application, Society for Industrial and Applied Mathematics (2000)

Abductive Validation of a Power-Grid Expert System Diagnoser

José Ferreira de Castro and Lu’s Moniz Pereira [*]

Centro de Intelig□ncia Artificial □CENTRIA
Universidade Nova de Lisboa
Monte da Caparica
2889-156 Caparica, Portugal
jeacastro@mail.telepac.pt
lmp@di.fct.unl.pt

Abstract. Transportation of electrical energy is normally made through a network of high-tension lines. In case of an incident, the electrical protections at both ends of a line are activated. Most of the incidents are short-lived, and can be solved by fast automatic reclosure of the breakers. When automatic reclosure fails, or does not happen, a timed reclosure is attempted after a few minutes, either manually or by automatic systems. When an incident occurs, several hundred messages sent by the substations can reach the control centers within a few seconds, making the human operator□s interpretation of the incident very difficult. The purpose of the previously developed SPARSE (Expert System for Incident Analysis and Power Restoration Assistance) is to assist the human operators in handling the emergency situations, giving them readable and accurate information. SPARSE is based on a logic programming inference engine reasoning over a set of time-stamped events. The new problem we addressed was how to validate the inference rules of SPARSE, by showing that no set of real events and diagnoses could be abduced that would violate the physical and logical integrity constraints of the problem domain, in order to certify the correctness of SPARSE with respect to the desired constraints. In this paper we examine how the sophisticated abductive logic programming system ABDUAL was employed for this purpose, and the practical tools developed and implemented to that end.

Keywords: Expert Systems

1 Introduction and Historical Overview

Transportation of electrical energy is normally made through a network of high-tension lines. In the case of a short-circuit in a line, the protection at both ends is activated. Natural causes (birds, lightning, storms, etc) can originate incidents that are short-lived and are immediately solved by the fast automatic closure of the breakers. When an automatic reclosure fails, or does not exist for that particular line, a timed reclosure is attempted after a few minutes, either manually or by automatic systems.

When a serious incident occurs, several hundred messages from the electrical substations, reporting the opening and closing of the circuit-breakers, can reach the

[*] Contact author

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 838-847, 2004.

control centers in a few seconds. The human operators can be flooded with raw information, quite difficult to read and interpret in real time. The purpose of SPARSE (Expert System for Incident Analysis and Power Restoration Assistance), based on logic programming, is to assist the human operators in handling the emergency situations, giving them readable and useful information.

One can find technical descriptions of SPARSE in the bibliography [1-9]. The system gathers basic information about the state of the network through a system called SCADA (Supervisory Control and Data Acquisition), and processes this information through a data-driven inference engine that offers in real-time the needed interpretation of what is going on. SCADA was first installed in the portuguese electrical network in 1988-89, and was totally remodeled in 2000. The SPARSE system, was actually used in one of the control centers in the north of Portugal, and is now being adapted to the new SCADA configuration. All comments made in this paper refer to the former version of SCADA.

Before being used in real-time mode, SPARSE was tested off-line using sample data gathered from real incidents. Some validation and verification tools were used to detect possible errors in the rule structure of the SPARSE inference engine, reported in [10-12]. But none of these tools can guarantee that an undesired conclusion may not appear due to an unforeseen configuration of the basic SCADA data.

The possibility of applying abduction techniques to SPARSE were studied as part of a master□s thesis in AI at Universidade Nova de Lisboa (UNL) in 2000 [14]. These techniques (using ABDUAL [13]) were developed at the Artificial Intelligence Centre (CENTRIA) of UNL, and a formal extended presentation (in Portuguese) can be found in the thesis, and are here reported for the first time in English.

The goal of applying abduction is twofold. First, we wish to identify all the possible hypothetical scenarios of the SCADA messages leading to a given conclusion of the SPARSE inference engine. This abductive reasoning allows us to verify that no unexpected and undesired sequences of events can bring about an untoward conclusion, one that violates physical and logical integrity constraints of the problem domain. Second, we wish to find the SCADA scenarios that are needed to achieve some given untoward conclusion. If the set of SPARSE inference rules is sound, these scenarios should be physically impossible to achieve, meaning there should be no realizable physical abductive explanation for them. Otherwise, the inference rules must be corrected on the basis of the abductive scenarios found.

The advantage of abductive reasoning over a set of tests with real data is obvious: it is hard to guarantee any set of tests is complete enough to cover all possible SCADA message configurations. Abduction, on the contrary, lists all possible explanations for a certain conclusion, being akin to model verification. This assurance is essential for the confidence of the final user in the SPARSE system. The results obtained showed the pertinence and usefulness of the abductive approach. The work done was of a very practical nature, building bridges between existing implemented systems, SPARSE on the one hand, and the logical programming abductive machinery on the other.

2 How SPARSE Works

To understand the challenges involved and the usefulness of the abductive approach, it is worth going a little into some of the details of SPARSE. The set of SPARSE rules used to test the abductive verification concept was a development version, because of copyright restrictions. This was actually convenient to verify the usefulness of abduction to detect potential problems in the version under development.

SPARSE is basically an inference engine reasoning over time-stamped events given by SCADA. For instance, SPARSE uses the time interval between the opening and closing of a circuit-breaker to conclude whether it was a manual or an automatic reclosure. A first production rule says that if the time interval is less than half a second, then the reclosure is automatic. Another production rule says that if the time interval is less than five seconds, then the reclosure is manual. Both rules are triggered by the same event - the opening of the circuit-breaker - and are scheduled to look for data and produce their respective conclusions after a certain amount of time compatible with the time conditions in the rules. In this case, a delay of half a second is necessary for the first rule, and of five seconds for the second. If the first rule succeeds, it takes out from the dynamic database the SCADA facts it used, and the second rule will therefore not succeed.

The main set of messages sent by SCADA is very simple: it gives information about the opening and closing of the circuit breakers, with a special separate message for the breakers’ sudden opening due to a surge of current (a short-circuit). SCADA sends also information about the status of the installations where the breakers are located: if they are in local or remote control mode, and in automatic or manual mode. These SCADA facts and the time conditions in the inference rules of SPARSE were chosen to become the abducibles (i.e. the basic facts for the abductive explanations).

Using the basic facts, SPARSE builds up a first level of very simple conclusions, and then uses the first level conclusions and other basic facts to build up more general conclusions of what is going on in the power grid. An important difficulty SPARSE has to deal with is the time attached to each SCADA event: it is not the time of the event itself, but the time when the SCADA information reaches the control centre. Because of unpredictable delays in the transmission, the recorded order of events can be different from what really happened. SPARSE needs to work on the basis of the time proximity of events (using often the module of time differences), but cannot reason with confidence over the order of events reaching the control centre in short intervals of time.

We saw above, in the example for automatic/manual reclosure, that each SPARSE rule is triggered by one of the events sent by SCADA. The actual activation of the rule and production of conclusions is generally delayed by a predefined time interval, in order to allow for the other relevant facts, listed in the body of the rule, to be recorded in the database. Two or more production rules can be triggered by the same event, with different activation delays. When a set of conclusions is produced by a rule that succeeds, its trigger and the other relevant facts to be retracted from the dynamic database are explicitly indicated in the head of the rule (actually they are changed to “old-facts” for possible analysis). Conclusions are also time-stamped, and are normally assigned the time of one of the SCADA facts used in the rule.

Rules using the same SCADA triggers to reach different conclusions will therefore be logically linked through the removal of data. In the above example, one can notice that the time conditions for manual or automatic reclosure are not exclusive. They overlap, and the SCADA programmer relied on the removal of data to avoid both rules succeeding. But since SCADA is continuously adding facts to the database, we may need physical considerations to get an assurance that no undesired facts are present when the rule for manual mode is activated. Let us suppose that the two rules for automatic/manual reclosure were triggered at time 0 by the opening of a circuit-breaker. A reclosure 0.3 seconds later would make succeed the rule for automatic reclosure (activated at time 0.5), erasing from the dynamic database the opening and closing information of the circuit-breaker. But we can still imagine a second short-circuit, say 4.6 seconds after the first one, followed by an automatic closure of the circuit-breaker at time 4.9. This would add to the database two SCADA facts allowing the rule for manual to succeed (activated at time 5.0), while the new rule for automatic (triggered at time 4.6 and activated at time 5.1) would fail. This problem could be solved by having SPARSE (a) check the correct time distance of the trigger found by the rule, or (b) by changing the time conditions of the rules to avoid the overlapping or (c) by checking the absence of a previous conclusion for automatic. These and many other even subtler complexities of SPARSE are typical of a program built over a dynamic database heavily using asserts and retracts.

3 Using ABDUAL

ABDUAL, an implemented abductive goal directed and tabled proof procedure under the well-founded semantics, needs a static set of rules to work with at any one time. A correct translation of SPARSE production rules into the adequate static rules is of the essence, and probably the most delicate and difficult step. How can we guarantee the translated static rules have the same meaning that the SPARSE rules have? In a direct translation we cannot, since there is vital information that cannot be found by a simple examination of the SPARSE rules. For instance:

a) The information about the delays and the triggers that activate the rules is not explicitly indicated within the rules. The resulting priorities among rules (which rule is activated first?) cannot therefore be subjected to an automated translation.
b) As we saw in the example above, rules are logically linked by the creation and deletion of dynamic variables.
c) Some SPARSE rules produce multiple conclusions from a given set of SCADA events. Not all events in the body of the rule are related to a given conclusion. For instance, an inference rule can produce the conclusion that a local manual closure was performed after a short-circuit that occurred in a substation. Since both conclusions (the opening due to the short-circuit and the manual closure) are produced by just one rule, abductive reasoning will be unable to distinguish the SCADA facts in the body of the rule pertaining to each conclusion. This needs to be understood and handled through splitting of the original rules.
d) Physical constraints do not appear anywhere in the rules. In the example above, a possible flaw in the rules was imagined supposing two incidents can occur in less

than five seconds for the same circuit-breaker. If this is physically impossible, the problem does not exist at all.

A possible alternative is to try to perfectly model the dynamic behavior of SPARSE, including the implicit physical constraints, and translating it into a static logical program well-suited for ABDUAL analysis. But the problem persists: how can we be sure our model reflects all the subtleties of SPARSE behavior? To achieve this assurance, we need a perfect understanding of the way SPARSE interacts with real situations. But if we reach a perfect understanding, we need no further analysis tools. A compromise to gradually reach a better understanding of SPARSE was made, using the following approach:

1. A program was developed to make a fast direct translation of the SPARSE rules into a format that could be utilized by ABDUAL.
2. ABUAL was used to rapidly calculate abductive scenarios for chosen conclusions.
3. The abductive scenarios were then examined, explained, and criticized. Some undesired aspects could be explained by the lack of information, while other aspects could be related to problems in the structure of the SPARSE rules.
4. Adequate modifications were introduced in the original SPARSE rules, and then back to step 1.

The advantage of this approach is that it was simpler to analyze and understand the ABDUAL output than to try to understand directly the SPARSE rules. We started with simple subsets of SPARSE rules, and gradually enlarged this set. An example of a SPARSE rule, together with its direct translation, is given in Appendix A. An example of ABUAL output is also given in Appendix A. For larger sets of rules, this output can become fairly large and hard to read. Some tools were developed in order to quickly analyze the abductive explanations. E.g., we can find answers to the following question: What is the simplest explanation? Are there elements common to all explanations? Are there explanations that are just extensions of simpler ones? Etc.

The production rules of SPARSE often present the chicken-egg type of circularities. For instance, to produce the conclusion that an installation shifted to manual mode, a previous conclusion that it had shifted before to automatic mode must be present in the database. Conversely, to produce the conclusion that an installation shifted to automatic mode, a previous conclusion that it had shifted to manual mode must be present in the database. When we apply abductive reasoning to this type of situations, we get an infinite regression of explanations (in this case simple WFS circularity detection is not enough because events are time-stamped and thus no actual circularity exists). It is nevertheless possible to obtain a satisfactory solution to this problem using an alphabetical similarity criterion of the abductive explanations to stop the generation of new steps (see the Appendix B for an example of how well-founded semantics deals with other circularities). One must be nevertheless very careful to avoid the undesired masking of explanations caused by similar events coming from different inference rules. Tagging the abducibles with a code that identifies the corresponding rule introducing each one solved this problem, and helped to trace the abductive chains of explanations.

4 Conclusions and Future Work

One could wonder about the usefulness of an abductive approach in this setting. We obviously need a good understanding of how SCADA and SPARSE work to analyze properly the abductive scenarios given by ABDUAL. When we modify the SPARSE rules we are already starting a debugging process, but we need to understand what we are doing. We initially took SPARSE as-is, without any purpose of improving the rules, but the study carried out suggests that ABDUAL can indeed be most useful in the development phase of this type of programs. We found that if the inference engine is designed having from the start abduction clarity and simplicity in mind, rules are simpler to understand and easier to verify. Since the confidence of the network operators in the correctness of the set of rules is paramount, abduction is a very useful and effective tool to increase the level of confidence in SPARSE for real-life situations.

The utilization of ABDUAL imposed some simple and obvious strategies to limit the number of explanations. Avoiding *not*□s in the body of the SPARSE rules is important, because the number of explanations for a certain fact to be absent is normally much larger than the number of explanations for a fact to be present. It is also better to avoid wherever possible the use of uninstantiated variables in the body of rules. Nevertheless, the implementation of ABDUAL that was employed in this study had a very interesting feature, □constructive negation□, which allows for uninstantiated variables under default negation. The basic idea is that the negation of a fact □f(c)□can be expressed by □f(V),V\=c□, to the effect that V is a variable whose values are distinct from □c□. Thus the goal □not f(V)□evaluates to □f(V),V\=c□, if the only negative instance for □f(V)□is □c□. Of course, constructive negation must deal with all the other more complex cases arising in a logic program, but that is not the subject of this presentation.

In conclusion, ABDUAL was put to use to detect specification inconsistencies in model-based diagnosis system for power grid failure. Abduction was employed to attempt to abduce hypothetical but physically possible events that might cause the diagnosis system to come up with a wrong diagnosis violating its specification constraints. The method is akin to the model verification stance: One strains to abduce a model, comprised of abduced physical events, which attempts to make the diagnostic program inconsistent. If this cannot be done, the power grid diagnoser can be certified correct.

The aim of our abductive application was indeed to certify that a given expert system diagnosis module was provably correct with respect to foreseen physical events. To wit, the diagnosis logic program was executed under ABDUAL in order to establish that no sequence of (abduced) physically coherent events (ie. monitoring messages) could be conducive to a diagnosis violating the (temporal) constraints expected of a sound diagnosis.

This proved to be feasible, though it required us to introduce a constructive negation implementation of ABDUAL, not reported here, because the abduced message events had to be time-stamped with temporally constrained conditions, but not anchored to specific temporal time-stamp constants, and occurred often under default negation literals to the effect that no supervening event occured in some relative time interval. The system, the application, and its use are described in detail in [14].

A number of open problems worthy of exploration remain in this class of problems, susceptible of furthering the use of the general abductive logic programming techniques brought to bear.

References

SPARSE System. There is a Web page dedicated to the training and information of power central operators, called Project Satoren. SPARSE is part of this project. The Web address is: http://www.cim.isep.ipp.pt/Projecto-SATOREN/. There one can find several technical reports, some of them related to SPARSE.

[1] Z.A.Vale, A.M.Moura,"An Expert System with Temporal Reasoning for Alarm Processing in Power System Control Centers", IEEE Transactions on Power Systems, vol. 8, No. 3, pp. 1307-1314, 1993.

[2] Z.A.Vale, A.M.Moura, M.F.Fernandes, A.Marques,"SPARSE- An Expert System for Alarm Processing and Operator Assistance in Substations Control Centers", Applied Computing Review, 2(2), pp.18-26, ACM Press, 1994.

[3] Z.A.Vale, M.Fernanda Fernandes, C.Rosado, A.Marques, C.Ramos, L.Faria, "Better KBS for Real-time Applications in Power System Control Centers: What can be learned by experience?", 1st Int. Conf. on Successes and Failures of Knowledge-Based Systems in Real-World Applications, Bangkok, 1996.

[4] Z.A.Vale, L.Faria, C.Ramos, M.F.Fernandes, A.Marques, "Towards More Intelligent and Adaptive User Interfaces for Control Center Applications", Int. Conf. on Intelligent Systems Applications to Power Systems (ISAP'96), Orlando, Florida, pp. 2-6, 1996.

[5] Z.A.Vale, C.Ramos, L.Faria, J.Santos, M.F.Fernandes, C.Rosado, A.Marques, "Knowledge-Based Systems for Power System Control Centers: Is Knowledge the Problem?", ISAP'97, pp. 231-235, 1997.

[6] Z.A.Vale, A.M.Moura, M.F.Fernandes, A.Marques, C.Rosado, C.Ramos, □SPARSE: An Intelligent Alarm Processor and Operator Assistant□, IEEE Expert, vol.12, no. 3, Special Track on AI Applications in the Electric Power Industry, pp. 86-93, 1997.

[7] Z.A.Vale, C.Ramos, A.Silva, L.Faria, J.Santos, L.M,Pinho, M.F.Fernandes, C.Rosado, A.Marques, "SPARSE □ An Expert System for Power System Control Center Operator Assistance and Training", 4th World Congress on Expert Systems □ Application of Advanced Information Technologies, ITESM, Mexico City, 1998.

[8] Z.A.Vale, C.Ramos, A.Silva, L.Faria, J.Santos, M.F.Fernandes, C.Rosado, A.Marques, "SOCRATES- AN INTEGRATED INTELLIGENT SYSTEM FOR POWER SYSTEM CONTROL CENTER OPERATOR ASSISTANCE AND TRAINING", IASTED Int. Conf. on AI and Soft Computing, Cancun, Mexico, 1998.

[9] Z.A.Vale, C.Ramos, L.Faria,"User Interfaces for Control Center Applications", The 1997 Int. Conf. on Intelligent Systems Applications To Power Systems (ISAP'97), Seoul, pp. 14-18, 6-10 July, 1997.

[10] J.Santos, L.Faria, C.Ramos, Z.Vale, A.Marques,"VERITAS□ A Verification Tool for Real-time Applications in Power System Control Centers", In Procs of the 12th Int. Florida AI Research Society (FLAIRS'99). 511-515. Orlando, Florida, 1999.

[11] J.Santos, C.Ramos, Z.Vale, A.Marques, "Validation and Verification of Knowledge-Based Systems for Power System Control Centres", In Procs of the European Symp. on Verification and Validation of Knowledge Based Systems (Eurovav'99), Oslo, 1999.

[12] Z.A.Vale, J.Santos, C.Ramos, M.F.Fernandes, C.Rosado, A.Marques,"SPARSE - A Prolog Based Application for the Portuguese Transmission Network: Verification and Validation", 5th Int. Conf. on the Practical Application of Prolog - PAP 97, pp. 291-310, London, 1997.

[13] J.J. Alferes, L.M. Pereira, T. Swift, □Abduction in Well-Founded and Generalized Stable Models Via Tabled Dual Programs□, Procs. 16th Int. Conf. on Logic Programming (ICLP'99), pp. 426-440, MIT Press, Las Cruces, New Mexico, 1999. http://centria.di.fct.unl.pt/~lmp/ publications/online-papers/iclp99.ps.gz

[14] J.F. Castro, Verifica□□o Abdutiva de Um Sistema de Diagn□stico Baseado em Regras, Master□s thesis in AI, Universidade Nova de Lisboa, 2000 http://www.cs.sunysb.edu/~tswift

Appendix A

We present here an example of a SPARSE rule, with its corresponding translation suitable for abduction. SPARSE rules are written using a specific grammar, developed for this purpose. The first line is a tag identifying the rule, with a short description of the type of event the rule detects. The next lines are separated by conjunctions (□and□) and disjunctions (□or□). This rule identifies an opening of a circuit breaker by manual remote control in two cases: First, if there is an opening of a circuit-breaker (identified by a breaker code □10□coming from SCADA) that cannot be associated with a short-circuit opening (identified by a trigger code □01□message from SCADA), because the module of the time interval is larger than 0.3 seconds; Second, if there was an opening of a circuit-breaker without a short-circuit opening being present in the database. The installation also needs to be in manual and remote mode. The breaker opening message is the trigger for the rule. Conclusions in the head of the rule are separated from the body of the rule by the symbol ==>. In this case, two messages are retracted (with the functor □retract-fact□), and a new fact is added (with the □create-fact□ functor). Important new facts are also displayed on the control centre monitor, using the functor □write_message1□.

```
 rule j2 : 'OPENING OF BREAKER BY REMOTE CONTROL' :       [[
  message(Date1,Hour1,[Inst1,Panel1,[Inst2,NL,'BREAKER']],'BREAKER',
          '10') at T1
      and
message(Date2,Hour2,[Inst1,Panel1,[Inst2,NL]],'>>>TRIGGERED',
          '01') at  T2
      and
  condition(mod_diff_times_greater(T2,T1,30))
      and
  manual(_,_,Inst1) at T3
      and
  remote(_,_,Inst1) at T4]
      or
  [message(Date1,Hour1,[Inst1,Panel1,[Inst2,NL,'BREAKER']],'BREAKER',
           '10') at T1
      and
      not
        message(Date2,Hour2,[Inst1,Panel1,[Inst2,NL]],'>>TRIGGERED',
                   '01') at T2
      and
```

```
    manual(_,_,Inst1) at T3
       and
    remote(_,_,Inst1) at T4]]
       ==>
   [retract_fact(breaker(_,_,Inst1,Panel1,_,_,closed),_,T1),

create_fact(FactNo,breaker(Date1,Hour1,Inst1,Panel1,_,remotecontrol,
                opened),T1),
    write_message1(FactNo,Date1,Hour1,Inst1,Panel1,
                   [(200,'opening of breaker by remote control')]),
 retract_fact(message(Date1,Hour1,[Inst1,Panel1,[Inst2,NL,'BREAKER']],
                       'BREAKER','10'),T1,T1) ].
```

The automatic translation will prepare static rules (j2-1 and j2-2) with ABDUAL syntax for each disjunction and each newly created fact. In this case, only the second line in the head of the rule actually creates a new fact, which will be used by other rules (this line is indicated within the comment with □production 2□). Doing so, we can easily trace how the translation was achieved, and to what disjunction and head line it refers to. Retracted facts are ignored, since their relationship with other rules must be found using information not available at this level. The resulting translated rules are:

```
   /* j2-1 production 2 */
   fact(breaker(Inst1,Panel1,Inst2,remotecontrol,opened), T1)
    <-
   fact0(j2,1,2,message(Inst1,_Panel1,[Inst2,NL,breaker]],breaker,io),T
1,
   fact0(j2,1,2,message([Inst1,Panel1,[Inst2,NL]],triggered,oi),T2),
   condition(j2,1,2,mod_diff_times_greater(T2,T1,30)),
   fact(manual(Inst1),T3),
   fact(remote(Inst1),T4).

   /* j2-2 production 2 */
   fact(breaker(Inst1,Panel1,Inst2,remotecontrol,opened), T1)
    <-
   fact0(j2,2,2,message([Inst1,Panel1,[Inst2,NL,breaker]],breaker,io),T
1,
   not
fact0(j2,2,2,message([Inst1,Panel1,[Inst2,NL]],triggered,oi),T2),
   fact(manual(Inst1),T3),
   fact(remote(Inst1),T4).
```

From here onwards the ABDUAL implementation works in two steps: First it adds the corresponding dual rules to these translated rules; Then it allows the user to query the transformed program. A typical query would be:

```
   ?- absp(fact(breaker(inst1, panel1, i2, remotecontrol, opened),tc)).
```

This query sends the abductive solutions to a file that can be read by the user and further processed by the specifically developed analysis tools. Here are the basic solutions given:

```
fact0(g4,1,message([inst1,_h683,[_h684,_h685,breaker]],inst_in_command,oo),T0)
fact0(inic,1,manual(inst1),T1)
fact0(j2,2,message([inst1,panel1,[ _h687,_h688,breaker]],breaker,io),tc)
not fact0(j2,2,message([inst1,panel1,[_h687,_h688]],triggered,oi),T2)
```

The first two arguments in each fact indicate the rule and the disjunction inside the rule to which it is related. The other arguments describe the SCADA messages or time conditions that explain the SPARSE conclusion. In this solution we see that two other rules besides j2 are used in the explanation. The first line says the installation is in remote control, the second it is in manual mode, the third identifies the opening of the circuit-breaker, and the last indicates de absence of a short-circuit triggering the automatic opening of the circuit-breaker. In this case there was an incident in T1 that triggered the circuit-breaker, but too distant in time to be related to the opening in tc.

```
condition(j2,1,mod_diff_times_greater(T1,tc,30))
fact0(g4,1,message([inst1,_h775,[_h776,_h777,breaker]],inst_in_command,oo),T2)
fact0(inic,1,manual(inst1),T3)
fact0(j2,1,message([inst1,panel1,[_h779,_h780]],triggered,oi),T1)
fact0(j2,1,message([inst1,panel1,[_h779,_h780,breaker]],breaker,io),tc)
```

Appendix B

Circularities can be easily handled by ABDUAL, through working over the Well-Founded Semantics (WFS). Consider the following example:

abds([f/0]). f1 ← f, f2. f2 ← f, f1.

where f/0 is declared to be abducible. The queries ?- ab(f1,S) and ?- ab(f2,S) are both answered □no□ in the ABDUAL implementation. This is in agreement with WFS, because literals involved in positive loops are considered false. However, the query ?- ab(not f1,S) gives the following solution S = [ ] * [not f] + [ ]. It is not a minimal solution, in the sense that the positive circularity is a sufficient condition to impose not f1. But ABDUAL does not obtain only minimal abductive solutions, this being one of its strengths, since not always does one desire minimality, often one does not want to pay for minimality, or the number of solutions may be infinite. Also, if no abducibles are found or given, the complexity of ABDUAL remains polynomial, because it retains the complexity of WFS. See [14] for more elaborate examples.

Integrating Web Services and Agent Technology for E-learning Course Content Maintenance

Fuhua Lin and Lawrence Poon

Center for Computing and Information Systems,
Athabasca University, Athabasca, Alberta, T9S 3A3
oscarl@athabascau.ca
lawrence@3open.net
http://ccism.pc.athabascau.ca/html/personal/fuhualin.htm

Abstract. This paper describes an integrated system using autonomous agents and Web Services to maintain Web-based course materials. The system is to monitor the Web-based course materials and report its findings. A couple of agents are deployed in a distributed educational environment. One of the agents called Web Monitoring Agent is mainly responsible for detecting web hyperlink status and content changes. If it discovers that the content of the course materials of a particular course has significantly been updated or modified, it will trigger a Notification Agent to send out a message to the students taking the course, or to whosoever interested to receive the message. If the Web Monitoring Agent discovers broken hyperlinks, it will trigger the Notification Agent to send out a message to the course instructor to remedy the situations. The model and approach proposed can be generalized and used in Web-based information resources management systems.

Keywords: E-Learning, Autonomous Agent, Web Services

1 Introduction

One of the merits of Web-based e-Learning courses is that it can provide up-to-date information. In order to provide *current*, *correct*, and *complete* materials to students, course instructors need to update e-Learning course materials from time to time. The reasons for Web-based content maintenance are three-fold.

First, courses in the ever-changing field such as 'Computing and Information Systems', materials need to be updated more often than other courses. Working in such a dynamic distributed learning environment [1], course instructors or instructors often need to review and revise course materials in a short time frame.

Second, because of the complexity of the materials, and the short development cycles within which the materials are produced, our best human efforts are sometimes not adequate to prevent occasional errors from slipping through, and students should therefore be prepared to encounter the odd minor 'glitch' in online courses. However, course instructors should make the necessary adjustments for the benefit of students. Whenever there is a significant change in the content of designated web pages, stu-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 848-856, 2004.

dents who are interested in the topic and all students who are taking the course will be notified by the course instructor via e-mail.

Third, Web-based course materials have many hyperlinks. These hyperlinks need to be maintained regularly to ensure their availability. However, it is not uncommon for an online course to have a few hundreds of hyperlinks. These hyperlinks can be broken for many reasons, such as Web servers may be down because of hardware failure, Web pages may be relocated to another server, or power may be cut off in other part of the world. To maintain these hyperlinks solely by human efforts is becoming more difficult and time-consuming, if not impossible. The degree of difficulty is hard to comprehend if we consider the fact that hyperlinks can be 'dead' and 'alive' at different point of time. The need for an automated system to help course instructors to maintain hyperlinks is pressing.

Web Services are Internet-based application components with service-oriented architecture using standard interface description languages and uniform communication protocols. A Web service is any service that is available over the Internet, uses a standardized XML messaging system, and it is not tied to any one operating system or programming language [2].

In this research, an agent is a Java program that acts autonomously on behalf of a person or organization. The rationales for the agent-based approach are the followings. The most important feature to us is that conversation can take place directly between applications as easily as between Web browsers and servers.

First, it enables us to distribute tasks to numerous specialized, fine-grained components. This promotes the modularity and flexibility of systems, and the incrementality of system development. It lets new services come and go without disturbing the overall system. The agents have their local knowledge about specific tasks and their autonomy. Limiting the complexity of an individual agent simplifies *control*, promotes *reusability*, and provides a *framework for tackling interoperability*.

Second, because of their autonomous nature, users can adopt a 'fire-and-forget' approach. Users don not need to remember to invoke them explicitly at the right point of time; they are able to react for themselves if they have access to the right data. The central feature of software agents is the ability to independently carry out tasks delegated to them by people or other software. This reduces the workload of users. The ever-changing and distributed nature of both data (i.e. student information database, course instructor information database, and course materials database) and applications require that software not merely respond to requests for information but intelligently anticipate, adapt, and actively seek ways to support users. Presumably some agents could all run on a central server doing come labor-consuming tasks but some could be distributed to the users' computers. From a performance viewpoint, it is best for the students to run their personal agents to control and configure their agents' behaviors.

2 Related Work

A lot of agent applications for e-Learning have been studied and experimented these years. Paper [3] identifies the roles of agents in educational activities. Paper [4] identified three groups of intelligent agents for teaching and learning applications: digital classmate as a series of intelligent agents assisting students and performing tasks related to learning, digital teaching assistants as a series of intelligent agents assisting teachers and performing task related to instruction and course management, and digital secretary as a series of intelligent agents assisting all members of an educational community in performing various administrative assistance tasks. All three intelligent agents proposed in that paper are conceptualized as a web-based multimedia agent (character) communicating with their human clients using one or more communication channels, including: text, speech, voice recognition, and animated facial graphics. Each agent could include a series of independent or mobile agents assisting an educational group. Here we give some representative examples of agents in education. I-Help [5-6] is a working agent system developed by ARIES Lab (the Laboratory for Advanced Research in Intelligent Educational Systems) for peer help to university teaching, using a multi agent-architecture. I-help is a peer-help system designed to assist learners as they engage in authentic problem-solving activities. Thaiupathump *et al.* work investigated the effects of applying intelligent agent techniques to an online learning environment. They created the know-bots that automated the repetitive tasks of human facilitators in a series of online workshops. The findings indicated that the use of know-bots was positively associated with higher learner completion rates in the workshops [7]. Baylor defined three major educational potentials for agents as cognitive tools: (1) As assistants, managing information overload; (2) serving as a pedagogical expert; and (3) creating programming environment for the learner [8].

3 System Architecture

The system consists of a couple of web services located in different places. The web services include *notification web service*, *Web monitoring web service*, *student information web service*, and *instructor information web service*.

There are two types of task agents supporting the web services, Web Monitoring Agent and Notification Agent. As Web Services, they have a dual nature which combines both characteristics of Web Services technologies and Agent technologies: the abilities to be published, found and called as a service, and the ability to move from platform to platform and make autonomous decisions.

Web Monitoring Agent is to monitor targeted web pages and to determine whether or not the content in those pages have been significantly changed. The meaning of 'significantly changed' is based on a couple of pre-defined criteria. For examples, the number of hyperlinks or photos increased or decreased, or the content lengths of the web page by examining its MIME header. If it discovers such changes, it will trigger a Notification Agent to send out a message to those students who are interested to receive the message. Figure 1 shows the system architecture.

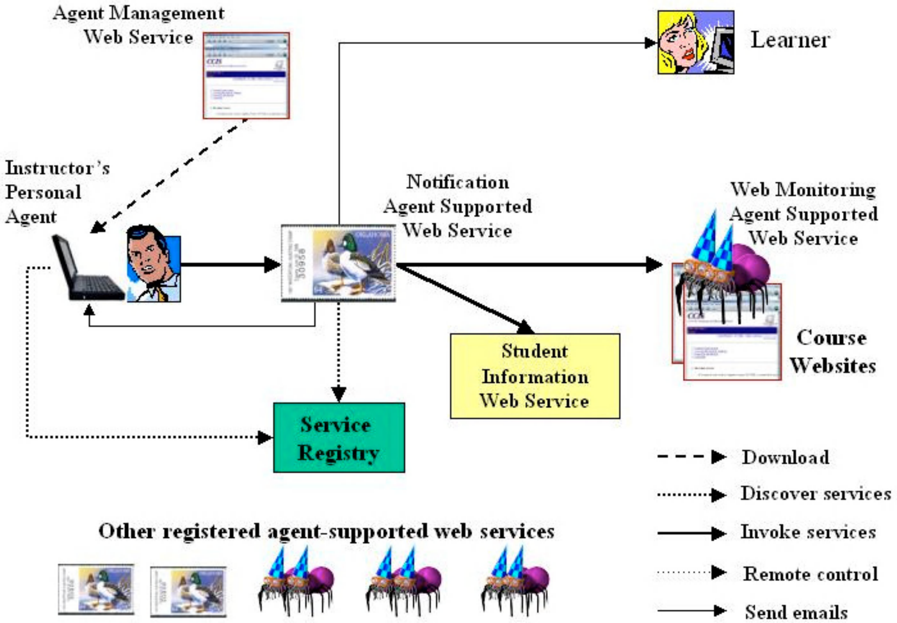

Agent Management Web Service
Learner
Instructor's Personal Agent
Notification Agent Supported Web Service
Web Monitoring Agent Supported Web Service
Course Websites
Student Information Web Service
Service Registry
Download
Discover services
Invoke services
Remote control
Send emails
Other registered agent-supported web services

Fig. 1. The System Architecture.

3.1 Agent Management Web Service

The Agent Management web service serves as a front end for agent management and deployment through Web technology. A registered user can login to download an agent platform and his/her favorite personal agent. Downloading can be through FTP or HTTP protocol. Once login in, a user can update his/her account information, or supply necessary information for agents to run. For example, a course instructor can provide his/her course information such as a course name and its base hyperlink, so that later a Notification Agent and a Web Monitoring Agent can make use of this information in order to process his/her request.

The Agent Management web service also acts as a proxy to an UDDI registry. It assigns unique agent identifications to agents and records agent information such as Agent Type, and relays them to an UDDI Registry. Personal Agent can search from the registry and invoke services provided by Agent Management Web Service. For example, Personal Agent can ask for the locations of Web Monitoring Agent who are free to work for the user.

The Agent Management web service is also a Web Services provider; it can serve SOAP-compliant clients by exchanging SOAP messages so that users can embed the results returned by the agents into their applications. For example, a course instructor can embed the broken hyperlinks, found by the Web Monitoring Agent, into his/her web email application and send them to students or build his/her course web pages with the logics taking care of the results returned by the Agent Management web service. Without coupling with Web Services, agents even though can notify the course instructor, there is a time gap between the broken links found and remedial actions taken. Coupling agents with Web Services, a course instructor has no urgent need to care about the remedial actions if contingent actions had already been taken.

3.2 Personal Agent

A Personal Agent, as a client of a service, it can perform searches of different entries stored in a UDDI. It then can make message and RPC style calls to a Web Service.

A Personal Agent is also an interface between the user and the multi-agent platform. Through the Personal Agent, a user can manipulate the options provided by other agents. For example, a course instructor can choose how often to receive email from the Notification Agent if broken links are found from his/her course materials. Personal Agent abides on the platform of the user's computer. Different group of users are assigned different types of Personal Agents. The assignment is based on their roles in the system. For example, the Personal Agents for course instructors are different from the students'. Course instructors can choose under what conditions they should be notified if the content of external links are changed. A Personal Agent is GUI driven and can be used to control all the agents with identification registered under the user name.

3.3 Web Monitoring Agent Supported Web Service

A Web Monitoring Agent has two functions, one is to detect broken links and the other is to detect web content changes. It scans the given pages periodically. When the agent detects a significant change (e.g. the link is broken), it sends a message to the Notification Agent.

Most of the work is done by agentized and multithreaded class *Spider* (www.JeffHeaton.com). A queue named *Workload* holds the base URL to be processed.

Step 1: A spider opens a connection to the base URL by openConnection().
Step 2: If the connection is failed, the whole process stops.
Step 3: If the connection is successful, the spider parses the web page to find all the URLs and put them into Workload.
Step 4: The spider checks other info from the MIME header such as Last Modified, Content Length etc., and stores them in the database for comparison purposes.
Step 5: Then the spider opens a connection to the next URL in work.
Step 6: If connection is failed, the spider will report this URL as broken links, store URL in database and open another URL in Workload.
Step 7: If connection is successful, the spider repeats step 3 to 6.
Step 8: The spider stops to work until no more URL in Workload.

3.4 Notification Agent Supported Web Service

Incorporating Notification Agents into the system is one of the ways for the multi-agent system to give responses to users. Notification Agent is responsible for sending out email on behalf of other agents in the multi-agent system. It is the postman of the whole community. Whenever an agent needs to send out email, it asks a Notification Agent to do so. The agent packages an agent message with the necessary details such as the message, the sender and the recipient email address and forward to the Notification Agent. Once received a message, the agent checks the validity of the information and sends the email out accordingly. The Notification Agent makes use of JavaMail class to perform the actual sending. The Notification Agent has no access to sender and recipient information, they are provided by other web services through XML request messages.

3.5 Student Information Web Service

The Student Information Web Service is designed to provide student information. For example, it maintains an email list of those students who are taking courses in open and distance learning environments of Athabasca University.

3.6 Databases

The database resource includes a student information database, an instructor information database, and a course link database. The simplified data model of the databases is shown in Figure 2.

4 Implementation

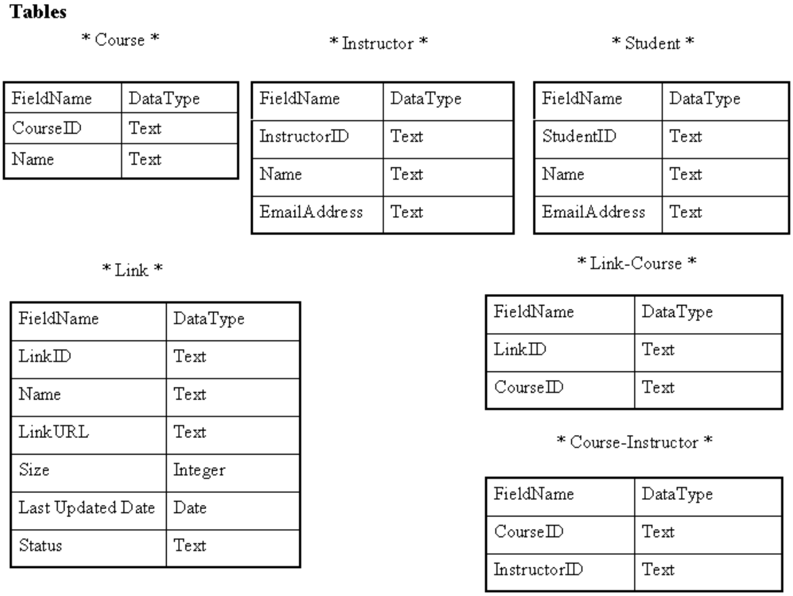
We implemented the agent system for the online course link maintenance using the architecture above. The agents and Web services are running on five different servers for testing purposes. The agents and the agent platforms are written in Java. We deployed the computers at different locations. The computers, Intel-based Pentium III class machines with 512MB RAM, are loaded with the following software:

Red Hat Linux 8.0
J2SE v 1.4.2
Apache Web Server 2.047 w/Axis 1.1
PHP 4.3.3
MySQL 4.0.14

From preliminary experimental results, the approach proposed is feasible. The Web services are provided by Apache Axis. We used JDBC to connect to MySQL databases.

Agent Communication
Agent talks to one another through SOAP messages. The SOAP messaging is provided by Apache Axis framework. For example, if a Personal Agent requests for bro-

ken link information from a Notification Agent, the Notification Agent will serve the Personal Agent by exposing its function <To Notify as Web Services>. The Personal Agent passes its unique identification number (a 16-digit alphanumeric number) and a request token (for example requestForBrokenLinkInfo) to the Notification Agent. The Notification Agent sends out the email and returns the request status to the Personal Agent according to the identity of the Personal Agent and the request token.



Tables

* Course *

FieldName	DataType
CourseID	Text
Name	Text

* Instructor *

FieldName	DataType
InstructorID	Text
Name	Text
EmailAddress	Text

* Student *

FieldName	DataType
StudentID	Text
Name	Text
EmailAddress	Text

* Link *

FieldName	DataType
LinkID	Text
Name	Text
LinkURL	Text
Size	Integer
Last Updated Date	Date
Status	Text

* Link-Course *

FieldName	DataType
LinkID	Text
CourseID	Text

* Course-Instructor *

FieldName	DataType
CourseID	Text
InstructorID	Text

Fig. 2. Tables for the database.

The following shows the agent communication mechanism of the Persona Agent:

```
try {
String endpoint = "http://www.os4schools.net:8080/axis/Notification.jws";
        // expose Notification Agent as a Web Service
Service service = new Service();
Call call = (Call) service.createCall();
call.setTargetEndpointAddress(new java.net.URL(endpoint));
call.setOperationName(new Qname("NotificationAgent", "toNotify"));
        // ask Notification Agent what to do --- toNotify
String = (String call.invoke(
          new Object[] {"gJuGoQ2bUNLkzZI0", "requestForBrokenInfo"});
           // Agent Identification Number and Request Token
System.out.println(ret);
}
catch (Exception e) {
   System.err.println(e.to.String());
}
```

The advantages of the SOAP-based agent communication are its easiness and language-independance. For example, instructors or students can use their own Personal Agent written in the languages they like. The limitations of the SOAP-based agent communication are: (1) the communications cannot be initiated from any agents. Only those agents with Web services capability can respond to request, (2) Communications are not encrypted.

We are doing experiments to test the scalability and usability of the system. The participants are CCIS students and course instructors, especially those enrolled in COMP200, COMP308, COMP378, COMP 489, and COMP 689. We will see if the instructors and students perceive a benefit from the agent system. We will send a questionnaire to COMP200, COMP308, COMP378, COMP 489, and COMP 689 and other courses instructors and students when they complete the courses. The questionnaire also includes a section requesting detailed feedback on each user's experience in the agent. The experiment will focus on the perceptions of the students regarding the helpfulness and overall usefulness of the agent system.

Perceived satisfaction will be measured by a questionnaire asking about the students' perceptions of the quality improvement of course materials in using the agent. The questionnaire will also be sent to course instructors and administrators to allow us to compare the work efficiency, i.e.

- how many broken links the agent detected,
- how much time between the course materials were changed and the students were notified of the change was shorten by using the agent system, and
- how much time of the course instructors were saved in maintaining course materials and notifying students and answering students' questions regarding course materials updates of the agent-supported content management system in comparison with non-agent course content management systems.

5 Conclusions and Future Work

We have described an agent system using autonomous agents and Web Services to maintain Web-based course materials. The integration of Web Services and Agent Technology is realized in the system in two ways. The agents can play the role of a Web Services client as well as the role of supporting Web Services. Personal Agent, a client of web services, can perform searching for different services stored in an UDDI registry. It then can invoke service calls to a Web Service. Notification Agent supported web service and Web Monitoring Agent supported web service have a dual nature which combines both characteristics of the two technologies: the abilities to be published, found and called as services, and the ability to make autonomous decisions. The model and approach proposed can be generalized and used in Web-based information resources management systems. For example, we plan to apply this approach to maintain eduSource Canada (http://www.edusource.ca/) learning object repositories.

Acknowledgements. We would like to thank Peter Holt and Steve Leung, who were involved with the original formation of this system. We also wish to thank the anonymous reviewers for their constructive comments.

This material is based upon work supported by Canada's NSERC discovery grant no.262147 and DALE research grant from the Athabasca University.

References

1. Alavi, M., Distributed Learning Environments, IEEE Computer, January 2004 (Vol. 37, No. 1), pp.121-122
2. Cerami, Ethan, *Web Services Essentials*, O'Reilly, 2002.
3. Lin, F., Holt, P., (2001). "Towards Agent-based Online Learning", *IASTED Int. Conf. Computer and Advanced Technology in Education (CATE),* June 27-29, Banff, Canada, pp.124-129.
4. Jafari, A., "Conceptualizing Intelligent Agents For Teaching and Learning*," International Conference on Intelligent Agents*, Las Vegas, USA, July 2001.
5. Greer J., McCalla G., Vassileva J., Deters R., Bull S., Kettel L. (2001) Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience, *Proceedings of AIED'2001*, San Antonio, 410-421.
6. Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A., and Vassileva, J.: The intelligent helpdesk: Supporting peer-help in a university course. In: Goettl, B. P., Halff, H. M., Redfield, C. L. and Shute, V. J. (eds.) *Intelligent Tutoring Systems. LNCS1452*. Springer-Verlag, (1998) 494-503
7. C. Thaiupathump, J. Bourne, J. O. Campbell, Intelligent Agents for Online Learning, *JALN* 3(2) - November 1999
8. Baylor, A. (1999). Intelligent agents as cognitive tools for education. *Educational Technology*, Volume XXXIX (2), 36-41.

Chemical Reaction Metaphor in Distributed Learning Environments

Hong Lin[1] and Chunsheng Yang[2]

[1]Department of Computer & Mathematical Sciences, University of Houston-Downtown
1 Main Street, Houston, Texas 77002, USA
linh@uhd.edu
[2] Institute for Information Technology, National Research Council
1200 Montreal Rd, Ottawa, Ontario, Canada K1A 0R6
Chunsheng.Yang@nrc.gc.ca

Abstract. This paper presents an application of Chemical Reaction Metaphor (CRM) in agent-based distributed learning systems. The suitability of using CRM to model multi-agent systems is justified by CRM's capacity in catching dynamic features of multi-agent systems in an e-learning environment. A case study in course material updating demonstrates how the CRM based language, Gamma language, can be used to specify the architecture of the learning environment. Finally, a discussion on the implementation of Gamma language in a distributed system is given.

Keywords: Agent-oriented modeling, e-learning, program specification, very high-level languages

1 Introduction

Distributed learning is becoming a more and more prevailing method for conveying courses in recent years. The absence of the needs for classrooms and fixed time schedule adds to the flexibility in course delivery, which includes: virtual classroom, asynchronous mode teaching, and mobility.

To allow these advantages, however, software engineering is burdened with unprecedented challenges of implementing such a learning environment, which should be of the following main features: adaptive curriculum sequencing, problem solving support, adaptive presentation, student model matching. It is very difficult to develop a system that could meet all requirements for every level of educational hierarchy since no single designer of such a complex system can have full knowledge and control of the system. In addition, these systems have to be scaleable and provide adequate quality of service support [1]. This gives reason to finding a model that can catch the interactive and dynamic nature of e-learning systems. Such a model should be general enough to address common architectural issues and not be specific to design issues of a particular system. A direct benefit of such a model is expressiveness and extensibility --- changes in the domain knowledge would not require an intensive system-wide modification to alter the information and all objects that initiate actions based on that changing information.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 857-866, 2004.

2 Agent-Oriented Modeling and the Chemical Reaction Metaphor

The agent concept provides a focal point for accountability and responsibility for coping with the complexity of software systems both during design and execution [2]. It is deemed that software engineering challenges in developing large scale distributed learning environment can be overcome by an agent-based approach [3]. In this approach, a distributed learning system can be modeled as a set of autonomous, cooperating agents that communicate intelligently with one another. As an example, Collaborative Agent System Architecture (CASA) [4, 5] is an open, flexible model designed to meet the requirements from the resource-oriented nature of distributed learning systems. In CASA, agents are software entities that pursue their objectives while taking into account the resources and skills available to them.

We found that the dynamic nature of distributed agents in e-learning environments makes it an ideal object for modeling by Gamma languages [6-9]. The concurrency and automation of agents require that the modeling language does not have any sequential bias and global control structure. In addition, the dynamic nature and non-determinism of interaction between an agent and its environment are suited to a computation model with a loose mechanism for specifying the underlying data structure. Therefore, chemical reaction metaphor provides a framework for the specification of the behavior of an agent. For example, data, which move around the internet, can be well modeled by chemical resolution; and mobile agents, which are created dynamically and transferred from clients to servers, can be included in the environment variable of a higher-order Gamma configuration. This provides a mechanism for describing both inter-agent communications and agent migration.

3 Specifying Multi-agent Systems in an E-learning Environment

From the workflow model of the course development, we can build a collaborative system model that partitions the problem into one or more smaller tasks, which are tackled by corresponding agents. For example, let’s examine the multi-agent system for course maintenance and recommendation that was designed in [10]. The online course materials are updated often in order to keep them as current as possible, esp. in some rapidly changing fields like ‘computing and information systems’. Because of the complexity of the materials, and the short development cycles within which they are produced, the course instructor should make the necessary adjustments time by time for the benefit of the students. Whenever there is a significant change on the content of several designated web pages of online course materials, students who take the course should be notified by the course coordinator by e-mail. Figure 1 shows the conversation schemata for course maintenance.

The conversation model of the course material change notification consists of the following elements. For simplicity of illustration, we assume that a student who takes the course is in one of the 3 phases, numbered 1, 2, or 3. The interpretation of the phases is trivial and left undefined (For example, phase 1 might be the phase before the first exam, phase 2 the phase between the first exam and the second exam; and phase 3 the phase between the second exam and the final exam.) except that we assume only students who have passed the previous phase are allowed to enter the next phase. A course web page also bears a phase number, indicating to which phase its content is significant. Once a change is made to a web page, all students taking the course and whose phase number matches the phase number borne by the web page will be sent the link to that page.

- ***Notification Agent Control Client (NACC)***: The Notification Agent Control Client of an instructor or a student runs on his/her machine and allows him/her to control the behavior of the corresponding Notification Agent deployed in a distributed environment. In our system, NACC adds a student into the student database or removes him/her from the database, or changes the phase number the student is currently in.
- ***Notification Agent (NTFC)***: The basic function of the Notification Agent is to send e-mails to students taking the course according to the student profiles stored in a database when the course material has been significantly changed.
- ***Monitoring Agent (MNTR)***: The Web Change Monitoring Agent of a system administrator monitors a collection of course material URLs stored in a database. When the agent detects a significant change, it sends a message to the Notification Agent. Also, once a broken link is detected in the topic tree, it notifies the maintenance agent to either correct the link or delete the orphaned page.
- ***Student Information Agent (STIF)***: A Student Information Agent is designed for providing services about student information, such as providing an e-mail list for a course by automatically maintaining the email list of students taking a course; and maintaining the profile of each student.
- ***Maintenance Agent (MNTN)***: The maintenance agent provides proxy services to the instructor. It maintains the content of the topic tree.
- ***Topic Tree or Link Database (LINK)***: The course material is organized in the form of a topic tree. Each entry in the topic tree is a link to a web page.

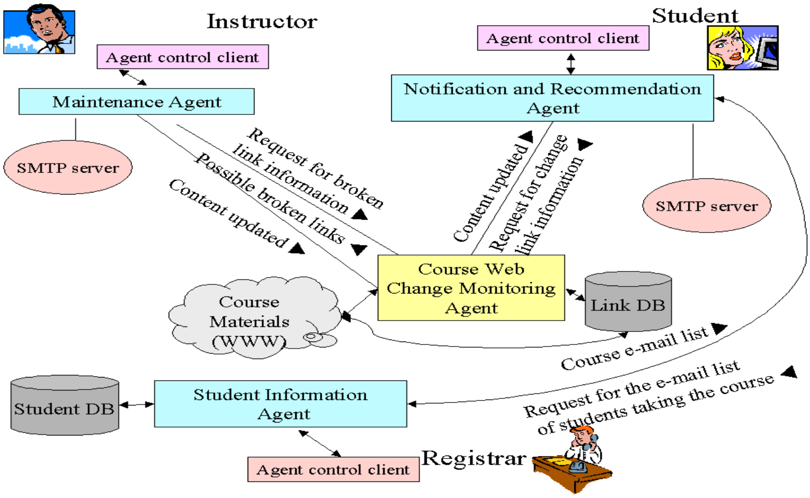


Fig. 1. A conversation schemata for course maintenance

Let *INST* and *STUD* denote the multisets of instructors and students, respectively, and *I*, *S*, and *L* denote the instructor (We assume that there is only one instructor), the initial roll of the class, and the initial content of the course (in the form of the set of links), respectively, the following is the Gamma program that specifies the above system:

MAIN i S0 L0 = [*P*, *NACC* = [*Q1*, *STUD* = *S0*], *NTFC* = [*Q2*, *STUD* = *S0*, *LINK* = *L0*], *MNTR* = [*Q3*, *LINK* = *L0*], *STIF* = [*Q4*, *STUD* = *S0*], *MNTN* = [*Q5*, *INST* = {*i*}, *LINK* = *L0*]] where

P = *P1* + *P2* + *P3* + *P4* + *P5*

P1 = [*Q1*, *STUD* = *S*+{(*s*, *1*, *Ø*)}]: *NACC*, [*Q2*, *STUD* = *S'*, *LINK* = *L*]: *NTFC* → [*Q1*, *STUD* = *S*+{(*s*, *1*, *Ø*)}]: *NACC*, [*Q2*, *STUD* = *S'*+{(*s*, *1*, *Ø*)}, *LINK* = *L*]: *NTFC* ← (*s*, *1*, *Ø*) ∉ *S'*

P2 = [*Q1*, *STUD* = *S*+{(*s*, **NULL**, *M*)}]: *NACC*, [*Q2*, *STUD* = *S'*, *LINK* = *L*]: *NTFC* → [*Q1*, *STUD* = *S*]: *NACC*, [*Q2*, *STUD* = *S'* - {(*s*, *p*, *M*)}, *LINK* = *L*]: *NTFC*

P3 = [*Q1*, *STUD* = *S*+{(*s*, *p*, *M*)}]: *NACC*, [*Q2*, *STUD* = *S'*+{(*s*, *p*, *M'*)}, *LINK* = *L*]: *NTFC* → [*Q1*, *STUD* = *S*+{(*s*, *p*, *M'*)}]: *NACC*, [*Q2*, *STUD* = *S'*+{(*s*, *p*, *M'*)}, *LINK* = *L*]: *NTFC* ← *M*   *M'*

P4 = [*Q2*, *STUD* = *S*, *LINK* = *L*+(*l*, *p*, **normal**)]: *NTFC*, [*Q3*, *LINK* = *L'*+{(*l*, *p*, **changed**)}]: *MNTR* → [*Q2*, *STUD* = *S*, *LINK* = *L*+(*l*, *p*, **changed**)]: *NTFC*, [*Q3*, *LINK* = *L'*+{(*l*, *p*, **changed**)}]: *MNTR*

P5 = [*Q1*, *STUD* = *S*+{(*s*, *1*, *Ø*)}]: *NACC*, [*Q4*, *STUD* = *S'*]: *STIF* → [*Q1*, *STUD* = *S*+{(*s*, *1*, *Ø*)}]: *NACC*, [*Q4*, *STUD* = *S'*+{(*s*, *1*, *Ø*)}]: *STIF* ← (*s*, *1*, *Ø*) ∉ *S'*

P6 = [*Q1*, *STUD* = *S*+{(*s*, **NULL**, *M*)}]: *NACC*, [*Q4*, *STUD* = *S'*]: *STIF* → [*Q1*, *STUD* = *S*]: *NACC*, [*Q4*, *STUD* = *S'* - {(*s*, *p*, *M*)}]: *STIF*

P7 = [*Q1*, *STUD* = *S*+{(*s*, *p*, *M*)}]: *NACC*, [*Q4*, *STUD* = *S'*+{(*s*, *p*+1, *M*)}]: *STIF* → [*Q1*, *STUD* = *S*+{(*s*, *p*+1, *M*)}]: *NACC*, [*Q4*, *STUD* = *S'*+{(*s*, *p*+1, *M'*)}]: *STIF*

P8 = [*Q2*, *STUD* = *S*+{(*s*, *p*, *M'*)}, *LINK* = *L*]: *NTFC*, [*Q4*, *STUD* = *S'*+{(*s*, *p*, *M*)}]: *STIF* → [*Q2*, *STUD* = *S*+{(*s*, *p*, *M'*)}, *LINK* = *L*]: *NTFC*, [*Q4*, *STUD* = *S'*+{(*s*, *p*, *M'*)}]: *STIF* ← *M*   *M'*

P9 = [*Q2*, *STUD* = *S*+{(*s*, *p*, *M'*)}, *LINK* = *L*]: *NTFC*, [*Q4*, *STUD* = *S'*+{(*s*, *p+1*, *M*)}]: *STIF* → [*Q2*, *STUD* = *S*+{(*s*, *p+1*, *M'*)}, *LINK* = *L*]: *NTFC*, [*Q4*, *STUD* = *S'*+{(*s*, *p+1*, *M*)}]: *STIF*

P10 = [*Q2*, *STUD* = *S*, *LINK* = *L*]: *NTFC*, [*Q5*, *INST* = *I*, *LINK* = *L'*]: *MNTN* → [*Q2*, *STUD* = *S*, *LINK* = *L'*]: *NTFC*, [*Q5*, *INST* = *I*, *LINK* = *L'*]: *MNTN* ← *L*   *L'*

P11 = [*Q3*, *LINK* = *L*]: *MNTR*,, [*Q5*, *INST* = *I*, *LINK* = *L'*]: *MNTN* → [*Q3*, *LINK* = *L'*]: *MNTR*, [*Q5*, *INST* = *I*, *LINK* = *L'*]: *MNTN* ← *L*   *L'*

Q1 = *Enrl* + *Drop*

Enrl =   (*s*, *1*, *Ø*): *STUD* ← *Enroll*(*s*)

Drop = (*s*, *p*, *M*): *STUD* → (s, **NULL**, *M*) ← *Drop*(*s*)

Q2 = *Updt* ∘ *Emal*

Emal = (*l*, *p*, **changed**): *LINK*, (*s*, *p*. *M*): *STUD* → (*l*, *p*, **changed**): *LINK*, (*s*, *p*. *M*+{*l*}): *STUD* ← *l* ∉ *M*

Updt = (*l*, *p*, **changed**): *LINK* → (*l*, *p*, **normal**): *LINK*

Q3 = (*l, p*, **normal**): *LINK* → (*l, p*, **changed**): *LINK* ← *Modified*(*I*)
Q4 = (*s, p, M*): *STUD* → (*s, p+1, M*): *STUD* ← *Pass*(*s, p*)
Q5 = *AddInst* + *AddLink* + *Chng* + *Updt*
    AddInst = *i: INST* ← *AddInst*(*i*)
    AddLink = *i: INST* → (*l, p*, **normal**): *LINK, i: INST* ← (*l, p*) = *AddLink*(*l, i*)
    Chng = (*l, p*, **normal**): *LINK, i: INST* → (*l', p*, **changed**): *LINK, i: INST* ← *l'*= *Change*(*l, i*)
    Updt = (*l, p*, **broken**): *LINK, i: INST* → (*l', p*, **normal**): *LINK, i: INST* ← *l'*= *Update*(*I, i*)

In this program, constants are written in boldface words. Each student record is a tuple (*student, phase, mailbox*) where *student* is the name of the student, *phase* the phase number where the student is in, and *mailbox* the mailbox of the student, which is a multiset of email messages. Each entry of the link database is also a tuple (*link, phase, status*) where *link* is the link to the web page in the topic tree, *phase* the phase number this page is designed for, and *status* the status of the page, which can be either **normal, changed, or broken**. Boolean functions *Enroll*(*s*) and *Drop*(*s*) return whether student *s* is enrolled in the class or wants to drop. *Modified*(*l*) function returns whether a particular web page pointed to by link *l* has been modified or not. *Pass*(*s, p*) function finds out whether student *s* has passed phase *p* or not. *Add*(*l, i*) function indicates whether instructor *i* wants to add page pointed to by link *l* into the link database or not. *Change*(*l, i*) function returns the link to the changed page whose original is pointed to by *l. Update*(*l, i*) function updates the broken link *l* and returns the corrected link.

The program consists of configurations in two levels: the *MAIN* configuration in the higher level and all other configurations in the lower level. Program *P* in *MAIN* configuration exchanges elements of the multisets in the environments of the lower-level configurations.

This example shows how Gamma language expresses the architecture of a multi-agent system succinctly. With the underlying computing model, we do not need to consider the specifications of nonessential features of the system, e.g., the number of program units, connection links for communications, and organizations of data, and therefore can focus on the specification of the overall architecture. It catches the way program units interact with one another and local computations, such as the implementations of those local functions, are left to the subsequent design phase.

The specification of the overall system benefits the subsequent design phases because details of the system can be added into the system in an accumulative fashion. The following section describes the specification of individual program units.

4 From Architecture to Building Blocks

A systematic design strategy was proposed in [11], in which Gamma specification of an agent system can be implemented in a hierarchical running environment composed of nodes in different levels of a tree. Interactions among agents can be implemented in a unified mechanism for synchronization. In this scheme, each configuration in the Gamma specification is implemented as a node. The overall architecture of the system is a tree structure, which expands and shrinks dynamically. A node only communicates with another node in the immediate upper or lower level. Interfaces between nodes specify the local conditions that may cause an action in the upper level. The actions in the upper level (in which nodes are called controlling nodes) can be creating/deleting nodes in the lower level or transforming the states of nodes in the lower level by data transfer.

The specification of the type of a node is composed of the module name, declarations of environment variables, imported variables, exported variables, and a body block consisting of sequentially executed statements.

```
process name(parameter-list)
    environment     Local environment variables
    import          Imported variables
    export          Exported variables
    begin
                    Statements
    End
```

Variables represent data sets. We leave the data structure for variables unspecified to maintain high-level abstraction. Imported variables and exported variables are written in the form of Module.Variable. If Module is omitted, the variable is identical to the local variable. Imported variables store values received from the nodes in the immediate lower or upper level while exported variables stores the values that are sent to the node in immediate lower or upper level. Both imported variables and exported variables can be interpreted as set of channels through which data are exchanged between nodes in adjacent levels. There is a separated channel established for each node. To maintain a high level of abstraction, we do not distinguish channels for different nodes. Instead, we use operator X.node to find out the sending node through variable X. Channels are automatic objects, which means imported channels receive messages whenever a send action is initiated by another node and exported channels send messages whenever data are available. To send a message to another node, we only need to use add action to add data items into the exported variables. Although we may use the same variable in both the import and export section, incoming data and outgoing data are distinguished by default. That means that outgoing data are never used in local computation.

Parameter list is used to pass initial values to the process when the process of the particular module is created. There are four actions that can be performed by a process:

- Add(variable, data): add data into variable
- Delete(variable, data): delete data from variable
- Select(variable): select an element of the data set represented by variable
- element.#n: projection operation --- extract the nth value of the tuple denoted by element

Statements in the body block of a module can be an add/delete action, a branching statement, or a looping statement. We omit the description of branching statements because they are not used in this program. The looping structure has the following syntax:

```
do   cond1 -> statement1;
     cond2 -> statement2;
     …
     condn: -> statementn;
od
```

The semantics of the looping statement is: cond1, …, condn are tested and one of the statements whose corresponding condition is tested to true is executed non-deterministically. Conditions are tested repeatedly until none of the conditions evaluates to true and the control is then transferred to the statement that follows the do statement.

The modules designed for the course maintenance program in the previous section is described in the following:

```
process NACC(STUD firstRoll)
environment
        STUD roll = firstRoll;
import
        STUD roll;
export
        STUD roll;
begin
        do   Enroll(s)       Add(roll, (s, 1, Ø));
             s = Select(roll), Drop(s)         Delete(roll, s), Add(roll, (s, NULL, s.#3));
        od
end

process NTFC(STUD firstRoll, LINK origLink)
   environment
        STUD roll = firstRoll; LINK link = origLink;
   import
        LINK link;
   export
        STUD roll; LINK link;
   begin
        do   l = Select(link), l.#3 = "changed" →
                  do s = Select(roll), l ∉ s.#3 → Add(s.#3, l);od
        do   l = Select(link), l.#3 = "changed" → Delete(link, l), Add(link, (l.#1, l.#2,
                  "normal"));
        od
   end

process MNTR(LINK origLink)
   environment
        LINK link = origLink;
   import
        LINK link;
   export
```

```
            LINK link;
    begin
            do   l = Select(link), Modified(l) → Delete(link, l), Add(link, (l.#1, l.#2, "changed"));
            od
    end

process STIF(STUD firstRoll)
    environment
            STUD roll = firstRoll;
    import
            STUD roll;
    export
            STUD roll;
    begin
            do   s = Select(roll), Pass(s.#1, s.#2) → Delete(roll, s), Add(link, (s.#1, s.#2 + 1,
                      s.#3));
            od
    end

process MNTN(INST initInst, LINK origLink)
    environment
            INST inst = initInst;
            LINK link = origLink;
    import
            LINK link;
    export
            LINK link;
    begin
            do   AddInst(i) → Add(inst, I);
                 i = Select(inst), l = AddLink(l, i) → Add(link, (l.#1, l.#2, "normal");
                 i = Select(inst), l = Select(link), l.#3 = "normal", l' = Change(l, i) →
                      Delete(link, l), Add(link, (l', l.#2, "changed"));
                 i = Select(inst), l = Select(link), l.#3 = "broken", l' = Update(l, i) →
                      Delete(link, l), Add(link, (l', l.#2, "normal"));
            od
    end

process MAIN(INST initInst, STUD firstRoll, LINK origLink)
    environment
            NACC nacc; NTFC ntfc; MNTR mntr; STIF stif; MNTN mntn;
    import
            STUD nacc.rollNacc, ntfc.rollNtfc, stif.rollStif;
            LINK ntfc.linkNtfc, mntr.linkMntr, mntn.linkMntn;
    export
            STUD nacc.rollNacc, ntfc.rollNtfc, stif.rollStif;
            LINK ntfc.linkNtfc, mntr.linkMntr, mntn.linkMntn;
    begin
            Add(nacc, NACC(firstRoll));
            Add(ntfc, NTFC(firstRoll, origLink));
            Add(mntr, MNTR(origLink));
            Add(stif, STIF(firstRoll));
            Add(mntn, MNTN(initInst, origLink));
    do   s = Select(nacc.rollNacc), s.#2 ≠ NULL, s ∉ ntfc.rollNtfc → Add(ntfc.rollNtfc,
                 s);
         s = Select(nacc.rollNacc), s.#2 = NULL, s’ = Select(ntfc.rollNtfc), s.#1 = s’.#1
                 → Delete(nacc.rollNacc, s), Delete(ntfc.rollNtfc, s’);
         s = Select(nacc.rollNacc), s’ = Select(ntfc.rollNtfc), s.#1 = s’.#1, s.#2 = s’.#2,
                 s.#3 ≠ s’.#3 → Delete(nacc.rollNacc, s), Add(nacc.rollNacc, s’);
         l = Select(ntfc.linkNtfc), l’ = Select(mntr.linkMntr), l.#1 = l’.#1, l.#2 = l’.#2,
                 s.#3 ≠ s’.#3 → Delete(ntfc.linkNtfc, l), Add(ntfc.linkNtfc, l’);
         s = Select(nacc.rollNacc), s.#2 ≠ NULL, s ∉ stif.rollStif → Add(stif.rollStif, s);
         s = Select(nacc.rollNacc), s.#2 = NULL, s’ = Select(stif.rollStif), s.#1 = s’.#1 →
                 Delete(nacc.rollNacc, s), Delete(stif.rollStif, s’);
         s = Select(nacc.rollNacc), s’ = Select(stif.rollStif), s.#1 = s’.#1, s.#2 + 1 = s’.#2,
                 s.#3 = s’.#3 → Delete(nacc.rollNacc, s), Add(nacc.rollNacc, s’);
         s = Select(ntfc.rollNtfc), s’ = Select(stif.rollStif), s.#1 = s’.#1, s.#2 = s’.#2, s.#3
                 ≠ s’.#3 → Delete(stif.rollStif, s’), Add(stif.rollStif, s);
         s = Select(ntfc.rollNtfc), s’ = Select(stif.rollStif), s.#1 = s’.#1, s.#2 + 1 = s’.#2,
                 → Delete(ntfc.rollNtfc, s), Add(ntfc.rollNtfc, (s.#1, s’.#2, s.#3));
         s = Select(ntfc.rollNtfc), s’ = Select(stif.rollStif), s.#1 = s’.#1, s.#2 + 1 = s’.#2,
                 → Delete(ntfc.rollNtfc, s), Add(ntfc.rollNtfc, (s.#1, s’.#2, s.#3));
         l = Select(ntfc.linkNtfc), l ∉ mntn.linkMntn → Delete(ntfc.linkNtfc, l);
         l = Select(mntn.linkMntn), l ∉ ntfc.linkNtfc → Add(ntfc.linkNtfc, l);
         l = Select(mntr.linkMntr), l ∉ mntn.linkMntn → Delete(mntr.linkMntr, l);
         l = Select(mntn.linkMntn), l ∉ mntr.linkMntr → Add(mntr.linkMntr, l)
    od
end
```

Note that higher-order operations remain in the module level. This makes the specification of the system closer to actual program. Also note that the transformation from Gamma specification to module specification can well be automated. Further transformation from module specification to programs in concrete language can be facilitated. The specification in the module level still focuses on generic process behavior. Data structures are left unspecified. Further refinement of the specification should include the use of data structures to organize the data sets. Therefore the Select operation can be implemented by an algorithm designed in accordance with the data structure. Another refinement would be the implementation of the data exchange channels.

5 Conclusions and Future Work

We propose a method for specifying a multi-agent system by using Gamma language. We find that in chemical reaction metaphor, architectural properties of a multi-agent system can be expressed succinctly and precisely. Through the case study, we demonstrate the usefulness of this method in the design of a multi-agent e-learning environment.We present a method for transforming the Gamma specification of the agent system into the specification in a module language, in which higher-order multiset operations are removed. This paves the way for implementing the specified system by using a sequence of program transformation. In the future, we will be working on the automation of the program transformation process and the refinement of module specifications by introducing data structures into the program.

References

1. Vouk, Mladen A., Donald L. Bitzer and Richard L. Klevans, Workflow and End-User Quality of Service Issues in Web-Based Education, IEEE Trans. on Knowledge and Data Engineering, 11, (4); July/August 1999, pp. 673-687
2. Yu, Eric, Agent-Oriented Modelling: Software Versus the World, Agent-Oriented Software Engineering AOSE-2001 Workshop Proceedings. LNCS 2222. Springer Verlag. 206-225.
3. Vassileva J., Deters R., Greer J., MaCalla G., Kumar V., Mudgal C., (1998) A Multi-Agent Architecture for Peer-Help in a University Course, Proceedings of the Workshop on Pedagogical Agents at ITS'98, San Antonio, Texas, 64-68
4. Flores, R.A., Kremer, R.C., & Norrie, D.H. An Architecture for Modeling Internet-based Collaborative Agent Systems, in T. Wagner & O.F. Rana (Eds.), Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, LNCS1887, Springer-Verlag, 2001, 56-63.
5. Lin F. O., Norrie D. H., Flores, R.A., & Kremer R.C. Incorporating Conversation Managers into Multi-agent Systems, in M. Greaves, F. Dignum, J. Bradshaw & B. Chaib-draa (Eds.), Proc. of the Workshop on Agent Communication and Languages, 4th Inter. Conf. on Autonomous Agents (Agents 2000), Barcelona, Spain, June, 3-7, 2000, pp. 1-9.
6. Banatre, J.-P., & Le Metayer, D. (1990). The Gamma model and its discipline of programming. Science of Computer Programming, 15, 55-77.
7. Banatre, J.-P., & Le Metayer, D. (1993). Programming by multiset transformation, CACM, 36(1), 98-111.
8. Banatre, J.-P., & Le Metayer, D. (1996). Gamma and the chemical reaction model: ten years after. in: Andresli, J.M., & Hankin, C. (eds.), Coordination Programming: Mechanisms, Models and Semantics, Imperial College Press.
9. Le Metayer, D. (1994). Higher-order multiset processing, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 18, 179-200.
10. Lin, F. O., Lin, H., & Holt, P., A Method for Implementing Distributed Learning Environments, Proc. 2003 Information Resources Management Association International Conference, May 18-21, 2003, Philadelphia, Pennsylvania, USA, 484-487.
11. Lin, H., A Language for Specifying Agent Systems in E-Learning Environments, in Fuhua Oscar Lin (eds.)*: Designing Distributed Learning Environments With Intelligent Software Agents*, Idea Group Inc., to appear.

An E-learning Support System Based on Qualitative Simulations for Assisting Consumers' Decision Making

Tokuro Matsuo, Takayuki Ito, and Toramatsu Shintani

Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso, Showa, Nagoya, Aichi, 466-8555, JAPAN.
{tmatsuo, itota, tora}@ics.nitech.ac.jp
http://www-toralab.ics.nitech.ac.jp/~{tmatsuo, itota, tora}

Abstract. In this paper, we propose an e-learning support system (LSDM) for assisting a buyers' decision making by applying artificial intelligence technology. When buyers purchase an expensive item, they must carefully select it from many alternatives. The learning support system provides useful information that helps consumers to purchase goods. We employed qualitative simulations because the result of output of simulation is useful. It consists of a qualitative processing system and a quantitative calculation system. When buyers use the system, they first input goods information they want to purchase. The information input by buyers is used in the qualitative simulation. Next, they fill out a form concerned with the details of their budgets, the rate of loans, and several other factors. After that, the system integrates the results of simulation and the buyer's input data and proposes plans to help their decision process. The system has several advantages: buyers can use it by simple input, they can understand process of simulation, and they can base their decision making on synthetic results.

1 Introduction

Services (e.g. e-commerce, e-learning, etc.) have made rapid progress in recent years, and there have been many investigations of the services. E-learning has been recognized as a promising field in which to apply artificial intelligence technologies [2][6][8][13]. Accordingly, we have developed an e-learning support system for assisting a buyers' decision making by applying artificial intelligence technology. When buyers purchase an expensive item, they must carefully select an item. In current study, we focus on the case that the buyer purchases an immovable item (e.g. house, land, etc.). When a buyer purchases such an item, they must take into account the price, the rate of a loan, their budget and several other factors. Furthermore, it is important to know whether the price will increase or decrease in the future.

There are several important factors concerned with the item's price, for example, an exchange rate, tax system, economic indicators, and fiscal policies. Some factors cannot be expressed as quantitative values. A qualitative method is therefore used to

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 867-875, 2004.

simulate a trend in price on the items. Such qualitative simulations output results as simple initial values. In our simulations, we can understand the mechanism and process of the simulations, because our qualitative methods consist of constructive graph models. On the model, relative factors are connected as a graph. In the simulation, nodes' conditions on the graph are changed as time passes. Our system also uses the quantitative calculation values concerned with total payments and so on. It can provide information in which buyers can trade virtually. By using our system, users can train how they should do the decision making when they want to purchase an expensive item.

The rest of the paper is organized as follows. Section 2 describes the outline of our user support system. In Section 3, we give some definitions and assumptions for the simulation. In Section 4, we show an example of qualitative simulations using our system. Finally in Section 5, we provide some final remarks.

2 The LSDM

In this section, we show an outline of our e-learning support system for users decision making (LSDM). The system consists of a qualitative simulation module and Quantitative calculation module. We propose a support system for integrating both modules. The qualitative simulation uses qualitative methods in which calculating values are classified some cases, e.g., "+", "0", "-", and so on. The quantitative module is a numerical calculation using the formulas. Our system provides the integrated results and users can learn how or when they should purchase (expensive) immovable items.

In the following, first, the system is outlined and the goal of our research is explained. Second, the qualitative simulation is demonstrated. After that, the numerical calculation is performed.

2.1 Outline of the LSDM

There are a number of researches concerned with e-services using artificial intelligence technologies [4][5]. We focus on an e-learning support system applying artificial intelligence technology when users want to purchase goods [10]. The main goal of the system is to give buyers a plan of the state on their total payments and forecasts of the price and loan rates. In recent years, there have been several studies about forecasting of stock dealings and financial dealing. Most of these studies employ quantitative methods using certain complicated formulas. These studies have been developed as practical methods, but it is difficult for non-specialists to understand their mechanisms and the meaning of their calculations. The LSDM system provides some information for general users how and when they should purchase goods.

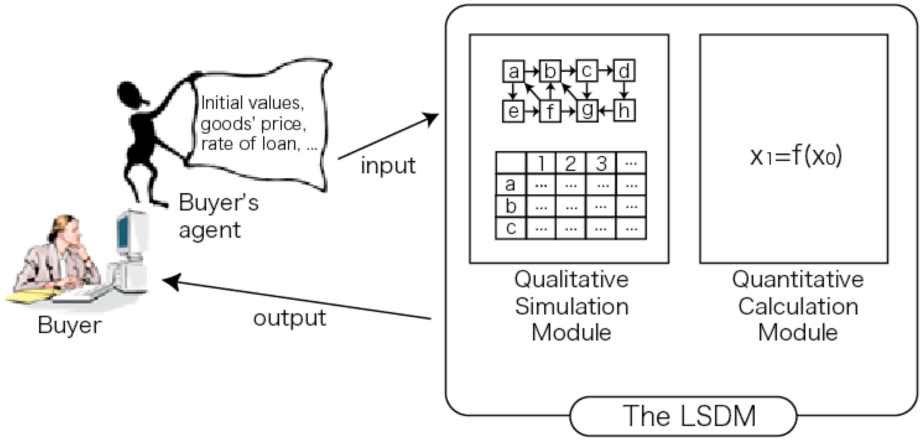


Fig. 1. Outline of LSDM

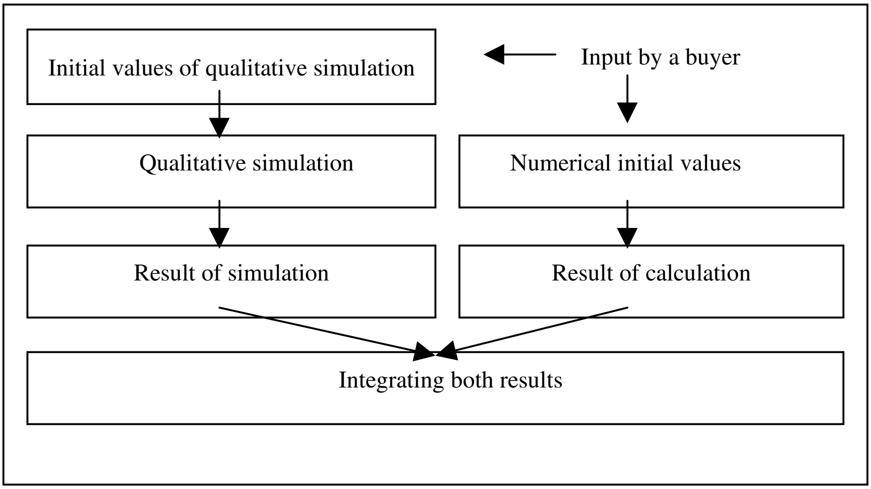


Fig. 2. Protocol of LSDM

Figure 1 shows a visual outline of the LSDM. First, a buyer inputs an initial value for the simulation, that is, a good's price, loan rates, total savings, and so on. Based on the buyer's input, their modules simulate a trend of item's price and calculate the total payment, after that the result of simulation and calculation are integrated and given to users. Finally, the LSDM shows the plans to purchase for assisting the buyer's decision making. The simulation and calculation in the LSDM are based on the protocol shown in Figure 2.

2.2 Qualitative Simulation

There are several methods for analyzing complex situations and the relations between cause and effect [1][6][11]. For example, a causal model using a directed graph is useful for analysis of complex situation, and we can observe a behavior of dynamics in the system [7]. In this research, we developed an e-learning support system for assisting a buyer's decision making when making a major purchase (such as a house). Volatilities of immovable goods have many factors. Table 1 lists some examples of the factors that consist of directs and indirects. The direct factors can be described in terms of the price of an immovable item, the loan rate, and total amount of a user's savings. The indirect factors are exchange rate, tax systems, business conditions, the financing system for real estate properties (e.g., housing loan), and several others. Some factors can be calculated using numerical methods, but others cannot be calculated based on quantitative methods.

Table 1. An example of factors

Direct factors	Price of immovable goods (quantitative) Loan rate (quantitative) Savings (quantitative)
Indirect factors	Exchange rates (quantitative) Tax system (qualitative) Business conditions (qualitative) Financing system (qualitative) and ….

It is difficult to forecast the price of an immovable item, and its price trend in the future, by using quantitative (numerical) methods. Accordingly, LSDM employs a qualitative simulation by applying qualitative reasoning through artificial-intelligence technology. An example of this qualitative simulation is given in section 3.

2.3 Numerical Calculation System

In the module for the numerical calculation, several mathematical formulas are used to determine the volatility of an item's price, rates of loans and a buyer's savings. These factors are important for forecasting a state in the future. Further, it is important for users to understand how much they must pay totally and how they plan to pay.

In the numerical calculation, first, one of the most important factors is the amount of user has saved and the price of the immovable item. When the savings are more than the price, a user can purchase the goods without price forecasts. When the saving is less than the price, a user must consider how to pay the money by using a loan. The next, important factor is this rate of the loan. Here it is assumed that the interest rate is fixed. When the rate is low, the total payment will be roughly equal to the one –off purchase price (i.e., the case of no loan). But if the rate is high and the balance is large, the buyer must pay a large amount of total interest due. Thus, the situation is complex depending on the terms.

A simple example of a general formula used to calculate the total cost of a housing loan. Principal G0 is defined as G0=G, where G is a total loan amount. Principal G1 is defined as G1=(1+r) G0–X=(1+r) G–X, where the 1st term, r, represents the loan rate and X represents the number of divided repayments. Generally, we can conduct the following formula in the *i*th term.

$$G_i = (1+r)^i G - \frac{(1+r)^i - 1}{r} X$$

It is assumed that the buyer pays in N installments. This formula shows that total repayment GN is zero after the Nth term. Total payment P given by the following formula.

$$P = \frac{rG}{1 - \frac{1}{(1+r)^N}}$$

The LSDM provides information with buyers as a set of qualitative and quantitative simulated data. Buyers can recognize the situation of the purchasing, and they can have a chance to purchase by appropriate strategy and decision making.

3 Simulation Primer

The simulation primer uses a causal model between causes and effects expressed as a trend graph [9]. Each node of the graph has a qualitative state value, and each arc of the graph shows a trend in effects. The characteristics of the nodes and arcs are explained in this section.

3.1 Qualitative States on Nodes

Each node has a qualitative state as time passes. We give three sorts of qualitative state values on nodes.

Def. 1 Qualitative State of Factors
The qualitative state [x(t)] is defined as given in Table 2. (node x at time t.)

Table 2. Qualitative states

[x(t)]	Qualitative states
H	LOW: In the next step, [x(t)] is not lower than in the current step.
M	BOTH: In the next step, [x(t)] is lower or higher than in the current step.
L	HIGH: In the next step, [x(t)] is not higher than in the current step.

3.2 State Trends Changing on Nodes

We define state trends changing on nodes that indicates differential on time. Three sorts of qualitative values are given.

Def. 2 Changing Trends of Nodes
The qualitative changing state [dx(t)] is defined as given in Table 3. (node x at time t.)

Table 3. Qualitative changing states

[dx(t)]	Qualitative changing state
I	[x(t)] is increasing.
S	[x(t)] is stable.
D	[x(t)] is decreasing

3.3 Direction of Effects of Arcs

It is defined state trends changing on arcs. We show the direction that the effect nodes have influence from the cause nodes. Two sorts of qualitative values are given.

Def. 3 Direction of Effects
D(x, y) is the direction of the effects from node x to node y as defined in Table 4. The directions are classified in two categories.

Table 4. Direction of effects

D(x,y)	Direction of effects
+	When x’s state value increases, y’s state value also increases. / When x’s state value decreases, y’s state value also decrease.
-	When x’s state value decreases, y’s state value increases. / When x’s state value increases, y’s state value decreases.

3.4 Transmission Speed of Effects on Arcs

Here, we define transmission speed of effects from node x to node y. The definition of the transmission speed is different from Defs. 1 to 3 essentially. The speed depends on the causal model.

Def. 4 Transmission Speed
The transmission speed V(x, y) is classified in Table 5.

Table 5. Transmission speed

V(x, y)	Transmission speed
V0	Node x’s value gives an effect to node y’s value immediately.
V1	Node x’s value gives an effect to node y’s value slowly.
V?	The speed is unknown.

3.5 Integration of Multiple Effects on Nodes

An integration of multiple effects from nodes is defined as follows. Figure 3 shows an example of the integration. When there are multiple adjacent nodes connected to a node, the effects are defined by the following definition.

Def. 5 Integration
When multiple nodes are connected to a node, the trends of effects are defined in Table 6. In the table, "?" means that the trends are not defined.

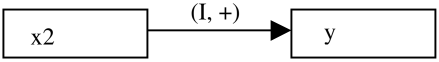


Fig. 3. An example of the integration

Table 5. Integration of state values

+	I	S	D
I	I	I	?
S	I	S	D
D	?	D	D

-	I	S	D
I	D	D	?
S	D	S	I
D	?	I	I

4 An Example of Qualitative Simulation

4.1 A Model for Qualitative Simulation

The causal/relation model for the qualitative simulation is explained in the followings. Figure 4 shows the causal/relation model for qualitatively forecasting the major purchase's price and the rate of a housing loan. Each arc has a characteristic (D(x, y) , V(x, y)). This model was constructed according to Japanese economical statistics from 1975 to 1990. It should be noted that the model is based on the following assumptions.

- When GDP increases, wages increase.
- When income increases, the amount of consumption increases.
- When the amount of exports increases, stocks inventories decrease.
- When GDP decreases, a tax reduction system is conducted (reduced).

4.2 An Example of Qualitative Simulation

Using the relation model and assumption given in section 3.2, we conducted an experiment on the trend in the price off a major purchase (a house). The initial values for the simulation are based on the economical conditions in Japan in 1975. Table 7 shows the initial values. In the simulation, when node's value cannot be determined by integration of multiple state values, the qualitative values are decided randomly.

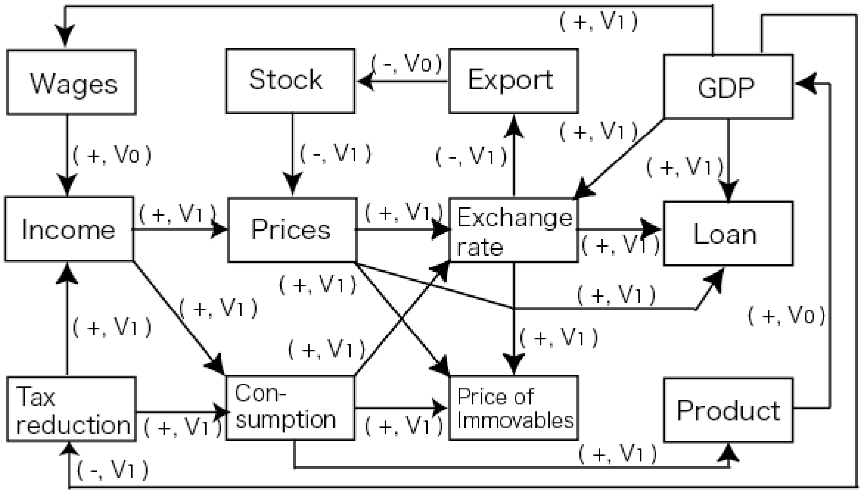


Fig. 4. A model off the relations between factors

Table 7. Initial Values

Trends	Factors(nodes)
Increasing	wage, income, price, consumption, export, exchange rate, stock, product, GDP, loan rate
Decreasing	tax reduction

The qualitative simulation is conducted in 200 time steps and the simulation results are shown in Figure 5. The horizontal axis represents time, while the vertical axis represents the price of immovable items.

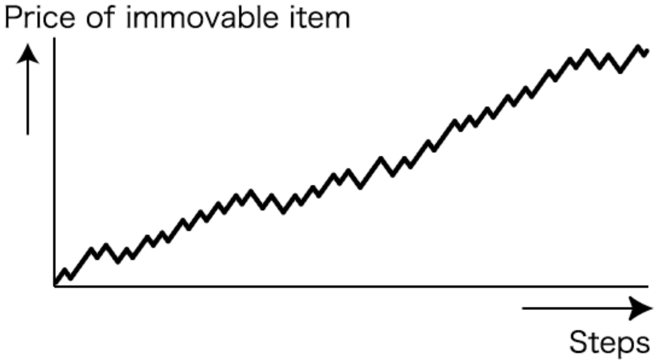


Fig. 5. Simulation results

5 Conclusion

In this paper, we proposed an e-learning support system (LSDM) for assisting a buyers' decision making by applying artificial intelligence technology. We employed qualitative simulations because the result of output of simulation is useful. The LSDM consists of a qualitative processing system and a quantitative calculation system. The system integrates the results of simulation and the buyer's input data and helps their decision process. The system has several advantages: buyers can use it by simple input, they can understand process of simulation, and they can base their decision making on synthetic results.

Our future work includes a development of LSDM to practice in the real world and apply to use as WBT on the Internet.

References

1. Agell, N., Aguado, C. J., A Hybrid Qualitative-Quantitative Classification Technique Applied to Aid Marketing Decisions, proc. of 11th International Workshop on Qualitative Reasoning, 2001.
2. Bredeweg, B., Forbus, K.: "Qualitative Modeling in Education", AI magazine, Vol. 24, No. 4, pp.35-46, American Association for Artificial Intelligence, 2004.
3. Chen, S. A., Wang, J., Yang, C. S., Constructing Internet Futures Exchange for Teaching Derivatives Trading in Financial Markets, proc of International Conference on Computers in Education, Vol.2, pp.1392-1395, 2002.
4. Forbus, K. D., Carney, K., Harris, R. and Sherin, B. L., A qualitative modeling environment for middle-school students: A progress report, International Workshop on Qualitative Reasoning, 2001.
5. Forbus, K. D., Helping Children Become Qualitative Modelers, Journal of the Japanese Society for Artificial Intelligence, Vol. 17, No. 4, pp.471-479, 2002.
6. Hata, S., Ohkawa, T., Komoda, N., Backward Simulation Method in Qualitative Simulation, Transaction in IEE Japan, Vol. 115-C, No.11, pp.1369-1376, 1995.
7. Kuipers, B., Qualitative Reasoning, The MIT Press, 1994.
8. Leelawong, K., Wang, Y., Biswas, G., Vye, N., Bransford, J. and Schwartz, D., Qualitative Reasoning Techniques to Support Learning by Teaching: The Teachable Agents Project, International Workshop on Qualitative Reasoning, 2001.
9. Matsuo, T., Ito, T., Shintani, T., A Structural Model for Qualitative Simulation-based a Contractors' Decision Support System, Proc. of 2003 Tokai-Section Joint Conference of the Institutes of Electrical and Related Engineers, p.251, 2003.
10. Matsuo, T., Ito, T., Hattori, H., Shintani, T., A Qualitative Simulation Model for Buyers' Decision Making, Proc. of the 20th Annual Conference of Japan Society for Software Science and Technology (JSSST), 2003.
11. Nishida, T., Qualitative Reasoning and its Application to Intelligent Problem Solving, IPSJ magasine, Vol.32, No.2, pp.105-117, 1991.
12. Russell, S., Norvig, P., Artificial Intelligence –A Modern Approach-, second edition, Pearson Education International, 1995.
13. Weir, R. S. G., The Rigours of On-Line Student Assessment – Lessons from E-Commerce, proc of International Conference on Computers in Education, Vol.2, pp.840-843, 2002.

Methodology for Graphic Redesign Applied to Textile and Tile Pattern Design

Francisco Albert[1], Jos□Mar'a Gomis[1], Margarita Valor[1], and Jos□Miguel Valiente[2]

[1] DEGI, Universidad Polit□cnica de Valencia, Camino de Vera s/n,
46022 Valencia, Spain
{fraalgil, jmgomis, mvalor}@degi.upv.es

[2] DISCA, Universidad Polit□cnica de Valencia, Camino de Vera s/n,
46022 Valencia, Spain
jvalient@disca.upv,es

Abstract. This paper presents a methodology for graphic pattern design and redesign applicable to tile and textile patterns. This methodology is used in a Design Information System whose reference framework is the scientific theory of symmetry groups. This Information System has two computer tools: one for the structural analysis of graphic designs and another interactive tool for the structural edition of patterns that, by means of the presented methodology, exploit all the capabilities provided by the manipulation of the minimum region (MR) of pattern designs. We present some application examples to generate new designs as modifications from designs acquired from historic sources. The methodology and tools are oriented to bridge the gap between the historical and artistic production of graphic design in the tile and textile industries.

1 Introduction

In the textile and tile industries Graphic Design is an essential element. For this reason, computer applications able to generate and edit decorative design patterns have been created. Furthermore the Design Departments count with different sources (ancient designs, catalogues, magazines and, in some cases, even specialized databases) of graphic information that help designers in their creative tasks. These facts, together with the reciprocity between pattern designs and the cultural environment [1] (the image of a product in many cases is associated with the region, history and agents involved in its manufacture) makes it seem unreasonable not to take advantage and capitalize the rich artistic, crafts and industrial heritage present in many regions by means of specific computer tools.

Nowadays, in our opinion, the lack of integration of the data sources in the design environment, at best generates attitudes in the designers less keen on using data sources than desirable, and in the worst case, makes the designer abandon decorative images existing in design data sources altogether. On the other hand, the lack of structural design tools for the exploration of these data sources has meant that the

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 876-885, 2004.

potential of Computer Aided Graphic Design has not been fully developed, affecting negatively the designer's activity.

We believe that the future trends of graphic design will come about through the integration of the designer's activity in a unique Computer Aided Design and Manufacturing Environment. This environment will have the following features:

1. Integration of the more conceptual stages of pattern design, using specific and user-friendly peripheral devices and developing interfaces that enable their use.
2. Creation and use of databases at the different stages of the design process. These databases should allow data consultation through content retrieval tools. Using the analogy of literature, these libraries will provide the designer with the existing vocabulary (motives) and lexicon (structures) for him to use in his own creations.
3. The development of advanced (structural) design tools for graphic design systems. Although object and basic pattern design creation and edition tools are commercially available in the field of Computer Aided Graphic Design, there are no interactive computer tools for advanced pattern design at structural level. The development of these tools will enhance the creative potential of designers.

2 Related Work

A large number of articles and books have been devoted to the study of ornamental patterns and tilings using the theoretical framework of the planar symmetry groups [1, 2, 3, 4, 5]. One of the first references is the system presented by Alexander in 1975 for generating the 17 symmetry patterns in plane [8]. Since then, many other research tools for automatic generation of symmetry patterns have been developed: Grünbaum and Shephard used a more sophisticated computer program to generate periodic tilings and patterns [2], Kaplan and Salesin developed a complete paremetrization for the isohedral tiles in order to given a closed figure in the plane, find a new closed figure that is similar to the original and tiles the plane without gaps nor overlappings [7].

Other authors have used simple geometric methods to generate islamic symmetric patterns [8] [9], mathematical approaches [10] [11] to generate abstract designs or L-systems [12] and similar approaches [13] to generate floral ornament

However they are mainly oriented to experiment with different algorithms for ornamental pattern creation. In order to analyze patterns we can find the works of Abas and Salman [14] and Ostromoukhov [15]. Both present the methods, but none has developed tools for automatic analysis.

Nowadays, we can find some commercial graphical applications that allow the edition of graphic designs using symmetries. Among them we can mention Terrazzo [16] included in Corel Photopaint and available as a plug-in for Adobe Photoshop, and SymmetryWorks [17] available as a plug-in for Adobe Illustrator. The major difference between them comes mainly from their raster (Terrazzo) or vector (SymmetryWorks) nature. Both select a unique MR for each plane symmetry group and directly apply the required isometries, but the edition process is restricted to the 17 symmetry groups without considering any other parameter.

As a summary we can conclude that, at this moment, no integrated graphic design application oriented to incorporate historical sources in the design cycle by means of ornamental design libraries with structural information based on the analysis of those sources, exist.

3 Information System for Graphic Design

Within this context, the authors of this work propose an Information System for Graphic Design (ISGD) [18]. This system uses the theory of symmetry groups [1, 2, 3, 4, 5] as reference framework, helping to develop a graphic design methodology for the textile and tile industries. This system has a modular architecture with two main modules: one consists mainly of two independent Databases: the Acquisition Database and the Pattern Design Database; and the other, implements two individual tools for Analysis and Edition. In this Information System the information flows as follows:

1. The acquired images are stored (together with related information) in the Acquisition Database.

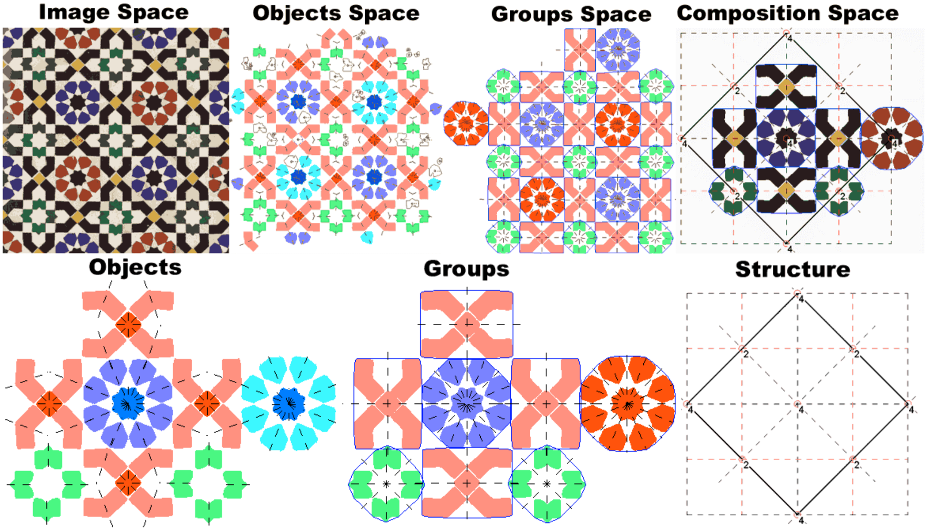


Fig. 1. Steps and products of the Analysis Tool

2. The Analysis Tool [19, 20, 21] performs a several step process (Fig. 1 up). In the process, the motives (objects and groups) are obtained in vectorial form with their symmetries and classifications. The FP (translations) and the plane symmetry group (rotations and symmetry axes) are also obtained. All this information (Fig. 1 bottom) with associated descriptors are stored in the Pattern Design Database.
3. The Edition Tool is an Adobe Illustrator experimental plug-in [22]. It has the purpose of adding to this program new edition methodologies (explained with more detail in the next sections) able to take advantage of the information stored in the Pattern Design Database. The new designs are also stored in this Database.

4 Methodology for Graphic Design and Redesing

We propose a design method that works at three levels: objects, groups and structure. Object and basic group edition are available in commercial graphic design applications. However the same does not happen with design composition structures. Furthermore the methodology incorporates the aspects indicated above, obtaining new designs from previously analyzed ones (redesign).

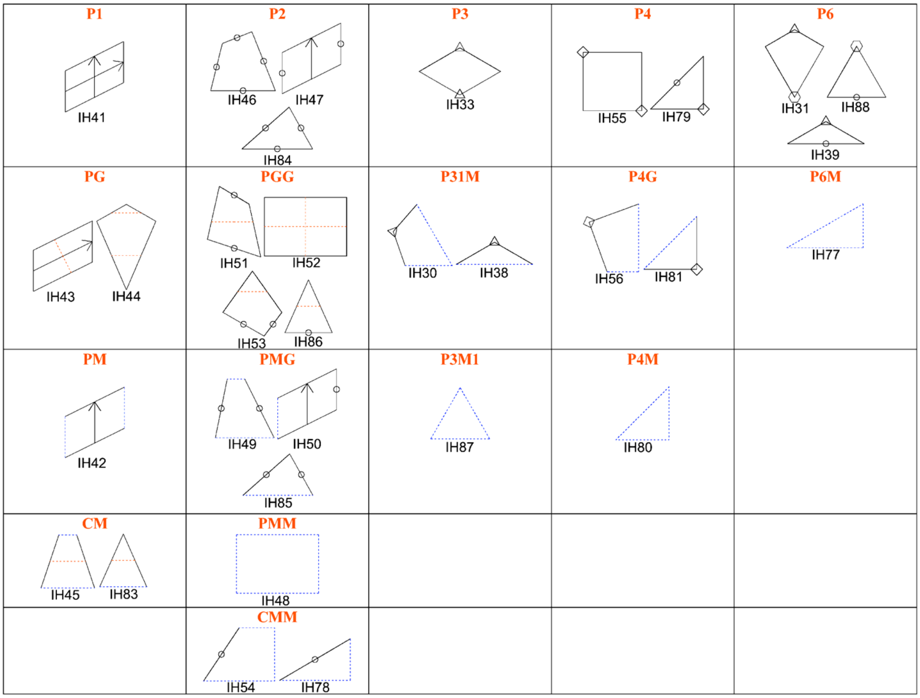


Fig. 2. Some MRs with 3 and 4 sides corresponding to the 17 plane symmetry groups

Our proposal for the structural edition of pattern designs is based on the fact that some small plane regions (minimum regions) are capable of generating different pattern designs depending on the isometries applied to fill up the plane.

According to the theory of symmetry groups, the design structures [5] (obtained with the analysis tool) are classified in 17 plane symmetry groups. Each one of them also has at least one MR (Fig. 2) [2] although some can have an infinite number of them, since the MRs geometry can have free parameters [7].

We should offer the designer the possibility of choosing between all the MRs available for each structure since, due to starting from different regions, the collection of variants generated will be different too. In addition, it will be necessary to know, for each MR, all the possible sets of isometries we can apply to it. So analysis of the MRs is required in two-ways:

1. It is necessary to organize the geometries based on its parameters. So that the strictest ones are related to the more generic ones (because they are considered special

cases of the generics). To the most restrictive, it will be able to apply the isometries of the generics ones, but not to the inverse ones.

2. It is necessary to determine the vertices of the MRs based on the vertices of the FP (obtained by the analysis tool) and the free parameters of the MR.

Also, this methodology has application for completely new designs [22]. In this case, the MR set is not determined by the plane symmetry group of an analyzed design, but so that we can choose any MR and modify its parameters freely. We can modify the sides of the MR so that they are not straight (the set of isometries is reduced to apply without gaps nor overlappings and less possibilities exist as with straight sides). The MR should be filled with motives from the database or created by the designer.

4.1 Geometry Diagrams

The MRs have been separated according to the number of sides (3, 4, 5 or 6) and four diagrams have been created organizing them according to their geometric restrictions [7]. The diagrams have been completed with the information necessary to generate all the variants possible. In Fig. 3 we can see the diagram of the MRs with four sides:

1. The shaded figures have indicated the parameters (v0, v1, v2 and v3) and the restrictions (the equal sides are indicated with equal number of perpendicular small lines; if they are parallel too, it is indicated with small chevrons. Between the sides that form 60°, 90° or 120° there are respectively, a small arc, a square or two arcs).
2. The lines between figures represent relations between restrictions: the most generic figures are located above and the most restrictive below (they are special cases of the superior ones, reason and for that the isometries of these can be applied to them). For the figures with four sides, the MR which can generate more variants (36) is the square, being the most generic since it is formed by four equal sides and four equal angles.
3. The non-shaded figures contain the isometries [2] to apply: displacements (arrows), rotations (180°, 120°, 90° and 60°, respectively, circle in half of side, triangle, square and hexagon in vertex), and symmetry axes (with sliding: dashed lines within the region; and without sliding: dashed lines in sides of the region).
4. The possible variants of the same plane symmetry group, generated by means of application of the same isometries but beginning by a different side or vertex, are indicated with the numbers of tiling (IH) greater than 100, 200 and 300. Sometimes it is possible to arrive at the variants from the same level (for example, the MR which generates the tiling IH50 has the variants IH150, IH250 and IH350). At other times it is possible to arrive only from more regular geometries (for example, the MR which generates the tiling IH53 cannot generate the variants IH253, IH153 and IH353, the first can be acceded from the geometries of the tiling IH44 and the two final ones only from the rhombus or inferiors).

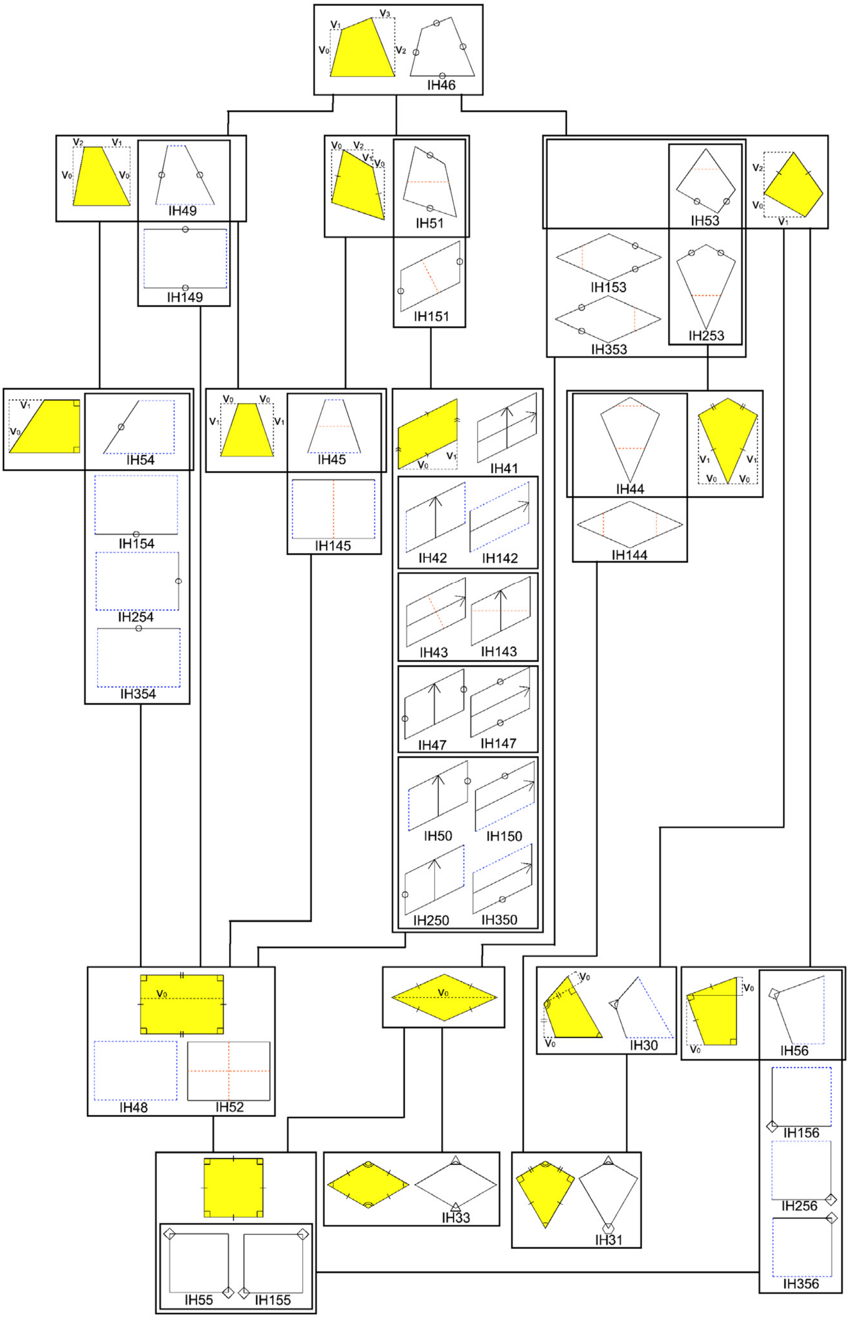


Fig. 3. Diagram of MRs with 4 sides

Fig. 3 does neither show numeration of vertices nor the requirements of each geometry to locate a concrete MR lower in the diagram in order to generate more variants.

4.2 Obtaining Minimum Region Vertices from the Structure

A geometry has been constructed for each MR. Next a motive was added and the corresponding isometries were applied to generate the pattern. A raster image was obtained to analyze him with the analysis tool. Fig. 4 shows an example. The origin (O) of the FP and vectors of its sides (L1 and L2) are known. The MR parameters v0 and v1 are shown too. The vertices of the MR are P1, P2, P3 and P4.

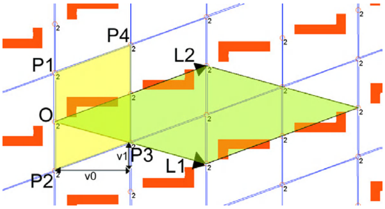


Fig. 4. Relationship between FP vertices and MR parameters with MR vertices

From a set of images for each MR, in which their parameters have been modified, the coordinates of the MR vertices have been obtained from the coordinates of the FP vertices and the MR parameters. From these images, we can reach two conclusions:

1. Not all the MR parameters (shaded figures in Fig. 3) are free an analyzed design since some of them belong to the FP and they would also modify the FP (for example, the parameters showed in Fig. 4). Nevertheless, if we are choosing a MR for a new design we will be able to modify them.
2. In some plane symmetry groups, there are other parameters (displacements and changes of orientation) that do not belong to the MR but to the FP, but they are free. We must consider them when we looked for MRs of an analyzed design, because they change the MR content (displacements) or even its form (changes of orientation).

5 Results

Fig. 5 shows a textile design. Its plane symmetry group is P1 since it has not symmetry axis nor centers of rotation common to all motives. We can see the FP (rectangle). The plane symmetry group P1 has a MR with four sides (corresponding to tiling IH41), with the same geometry as the FP. The restrictions indicate that its shape is a general parallelogram but, in this case, it is a rectangle (more restrictive) and we can locate the MR at an inferior level in the diagram of Fig. 3 (having 24 cases instead of 14).

The MR has two parameters which the FP fix, therefore we cannot change them. Nevertheless, the FP has two free displacement parameters in the direction of its sides. In the two right images of Fig. 5, we can see two possible MRs. In the right one a displacement has been carried out in order to avoid the edges from cutting the main motives (roses). This is the reason why this second option had been employed to generate the six designs of Fig. 6.

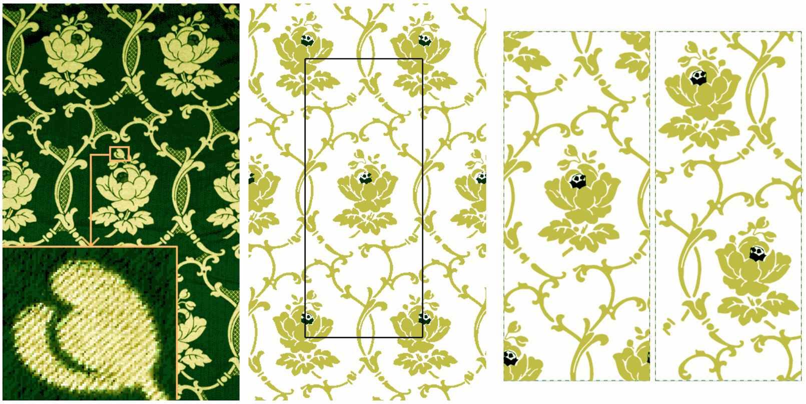


Fig. 5. From left to right: textile image with extended detail, design analyzed with FP rectangle and background removed and two possible RMs

Fig. 7 shows (left) a new design with discontinuities between regions, corrected by means of editing the MR motives (right). The whole design is automatically updated.

6 Conclusions

This paper presents a new methodology for graphic pattern redesign applicable to tile and textile patterns. The methodology combines the designer□s creativity with the use of historical data sources and is based on the theory of symmetry groups. In addition to this, it can be used to generate completely new designs.

The methodology works on the structural level, obtaining the MRs of analyzed designs and applying different sets of isometries to them. For this, the MRs have been separated according to the number of sides and four diagrams have been created organizing them according to their geometric restrictions; these diagrams contain the information necessary to generate all the possible variants from the first region.

This type of edition allows the designer to handle many new parameters in relation to the design structure (related to MR shape, tiling, FP shape or plane symmetry group), as well as the fast generation of families of designs.

The methodology has been implemented through an edition tool. This tool allows the presence of discontinuities editing only the MR motives to be solved, since the rest of the design is updated automatically. Our next target is developing interfaces using user-friendly peripheral devices.

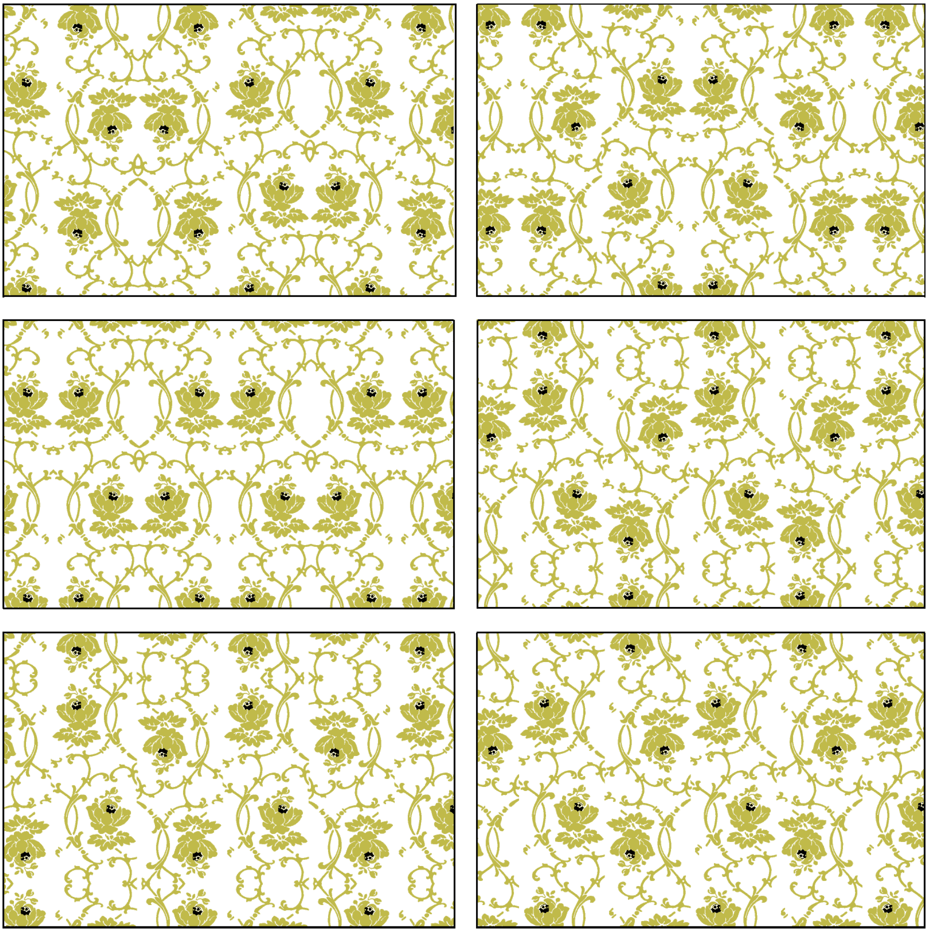


Fig. 6. Redesigns from the MR on the right of Fig. 5

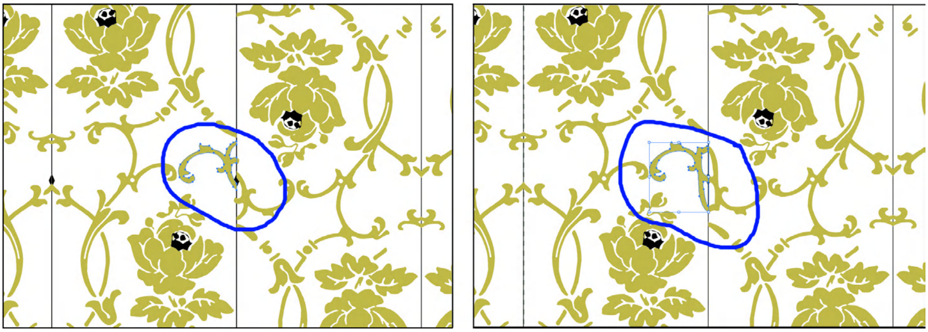


Fig. 7. Correction of discontinuities by means of MR motives edition

Acknowledgments. This work has been supported by the Spanish Science and Technology Ministry and the European Union (Project DPI2001-2713).

References

1. Washburn, D.K., Crowe, D.W.: Symmetries of Culture: Theory and Practice of Plane Pattern Analysis. University of Washington Press, Seattle (1988)
2. Gr□nbaum, B., Shephard, G.C.: Tilings and Patterns. W. H. Freeman, New York (1987)
3. Shubnikov, A.V., Koptsik, V.A.: Symmetry in Science and Art. Plenum Press, NY (1974)
4. Martin, G.E.: Transformation Geometry. An Introduction to Symmetry. Springer-Verlag, New York (1982)
5. Schattschneider, D.: The Plane Symmetry Groups: Their Recognition and Notation. The American Mathematical Monthly 85 (1978) 439-450
6. Alexander, H.: The computer/plotter and the 17 ornamental design types. Proceedings of SIGGRAPH□75, (1975) 160-177
7. Kaplan, C., Salesin, D.: Escherization. Proc. of SIGGRAPH 2000, (2000) 499-510
8. Abas, S.J., Salman, A.: Geometric and Group-theoretic Methods for Computer Graphic Studies of Islamic Symmetric Patterns. Computer Graphics Forum 11 (1992) 43-53
9. Kaplan, C.: Computer Generated Islamic Star Patterns. Visual Mathematics 2 (3) (2000)
10. Field, M.: Designer chaos. Computer-Aided Design 33 (2001) 349-365
11. Sen, A.K.: A Product-delay Algorithm for Graphic Design. Computers & Graphics 22 (1998) 759-764
12. Smith, A.R.: Plants, Fractals and Formal Languages. Proceedings of SIGGRAPH□84 (1984) 1-10
13. Wong, M.T., Zongker, D.E., Salesin, D.: Computer-Generated Floral Ornament. Proceedings of SIGGRAPH□98 (1998)
14. Abas, S.J., Salman, A.S.: Symmetries of Islamic Geometrical Patterns. World Scientific (1995)
15. Ostromoukhov, V.: Mathematical Tools for Computer-Generated Ornamental Patterns. Lecture Notes in Computer Science 1375 (1998) 192-223
16. Xaos Tools web site <http://www.xaostools.com/products/terrmain.html>
17. Artlandia web site <http://www.artlandia.com/products/symmetryworks>
18. Carretero, M., Valiente, J.M., Albert, F., Gomis, J.M.: Diseo□ de un Sistema de Informacin□ Gr□fica para Tareas de Rediseo□ Industrial. Pro ceedings of the XIII International Congress of Graphics Engineering, (2001)
19. Valiente, J.M., Albert, F., Gomis, J.M.: Feature Extraction and Classification of Textile Images: Towards a Design Information System. Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (2002) 77-94
20. Valor, M., Albert, F., Gomis, J.M., Contero, M.: Analysis Tool for Cataloguing Textile and Tile Pattern Designs. Proceedings of the II International Workshop on Computer Graphics and Geometric Modeling. (2003) 569-578
21. Valor, M., Albert, F., Gomis, J.M., Contero, M.: Textile and Tile Pattern Design Automatic Cataloguing Using Detection of the Plane Symmetry Group. Proceedings of the Computer Graphics International (2003) 112-119
22. Gomis, J.M., Valor, M., Albert, F., Contero, M.: Integrated System and Methodology for Supporting Textile and Tile Pattern Design. Proceedings of the III International Symposium on Smart Graphics. (2003) 69-78

Knowledge Representation on Design of Storm Drainage System

Kwokwing Chau and C.S. Cheung

Department of Civil and Structural Engineering, Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. During the design of storm drainage system, many decisions are involved on the basis of rules of thumb, heuristics, judgment, code of practice and previous experience of the designer. It is a suitable application field for application of the recent artificial intelligence technology. This paper presents the knowledge representation of the design of storm drainage system in a prototype knowledge-based system. Blackboard architecture with hybrid knowledge representation techniques including production rule system and object-oriented approach is adopted. Through custom-built interactive and user-friendly user interfaces, it furnishes designers with entailed expertise in this domain problem.

1 Introduction

In the past decade, the potential of artificial intelligence (AI) techniques for providing assistance in the solution of engineering problems has been recognized. A knowledge-based system (KBS) is considered suitable for solving problems that demand considerable expertise, judgment or rules of thumb. It has made widespread applications and is capable to accomplish a level of performance comparable to that of a human expert in different fields: vertical seawall design [1]; liquid retaining structure design [2-4]; site level facilities layout [5]; modeling in coastal processes [6]; flow and water quality modeling [7]; thrust block design [8]; fluvial hydrodynamics [9-11]; river flow routing [12].

In drainage engineering field, existing computer models may be available to perform a particular task in the whole design process [13-15]. It is difficult to code empirical rules or expert knowledge in a conventional algorithmic framework. Over-emphasis has been placed on algorithmic procedures in many computer-aided design packages, thus producing a large gap between model developers and users. This may produce inferior design and cause the under-utilization, or even total failure of these models. The application of these individual programs requires the intensive knowledge of the designer and they are prone to human errors during the data transferring processes. There is a need to develop programming environments that can incorporate engineering judgment along with algorithmic tool [16]. Yet, no attempt has been made to apply this intelligent system to this domain problem. Design of storm drainage system involves many decisions to be made by the designer based on

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 886-894, 2004.

rules of thumb, heuristics, judgment, code of practice and previous experience. Moreover, the conventional design refers to design charts [17] where pipe diameters are adjusted by trial and error, entailing laborious work. A need arises for an user-friendly computer design aid as well as training tool coupled with the automation of the design process to design the storm drainage systems speedily and precisely, and to avoid troublesome manual computations. KBS is suitable to furnish a solution to this decision making process through incorporating the symbolic knowledge processing.

This paper describes the knowledge representation of a prototype KBS for design of storm drainage system. This application domain has relations to the upcoming research area of "ecological informatics". The KBS developed is based on Civil Engineering Manual Volume IV: Sewerage and Drainage [18]. It is intended not only to emulate the reasoning process followed by drainage designers, but also to act as an intelligent tool that furnishes expert advice to its users regarding the design process and selection of design parameters. Its knowledge base comprises representations of the design entities, as well as the design knowledge of human experts. Through custom-built interactive graphical user interfaces, the user is directed throughout the design process.

2 Design of Storm Drainage System

The design of the storm drainage system is multi-disciplinary involving expert knowledge of different fields. The Colebrook-White Equation is used to describe the whole range of pipe flow and to calculate the velocity and hence the capacity. The choice of designed rainfall intensity depends upon the duration of a rainstorm and the adopted statistical frequency of recurrence. Rainfall intensity curves are acquired based on the statistical analysis of long-term rainfall records from the Royal Observatory of Hong Kong. The Rational Method is employed for estimation of design peak runoff to be conveyed in the storm drainage system of Hong Kong. The pipe size is first selected for the accumulated runoff at each manhole inlet. The minimum and maximum actual flow velocities are limited in compliance with the requirement of the Civil Engineering Manual. Besides, in the backwater checking computational procedure, adjustment of the diameter of the pipes at manholes are required to ensure the backwater levels are always lower than the finished levels and flooding will not occur. Only commercially available pipe diameters are allowed.

3 The Prototype System

A prototype KBS has been customarily designed such that the user can always overrule any design options and recommendations provided by the system. It allows for users to fine-tune the system to their preference style, yet to let them benefit from its thoroughness, speed, and overall effectiveness for designing storm drainage system. Although this system is tailored for design based on the rainfall intensity-duration-intensity curves in Hong Kong, it can be readily adapted to cater for other conditions in other countries. Most engineers can easily exploit the full potential of the system with minimum supervision and within a short time span. This system has been implemented with the aid of a microcomputer shell Visual Rule Studio, which is a hybrid application development tool under object-oriented design environment. Visual Rule Studio acts as an ActiveX Designer under the Microsoft Visual Basic 6.0 programming environment. Production rules as well as procedural methods are used to represent heuristic and standard engineering design knowledge.

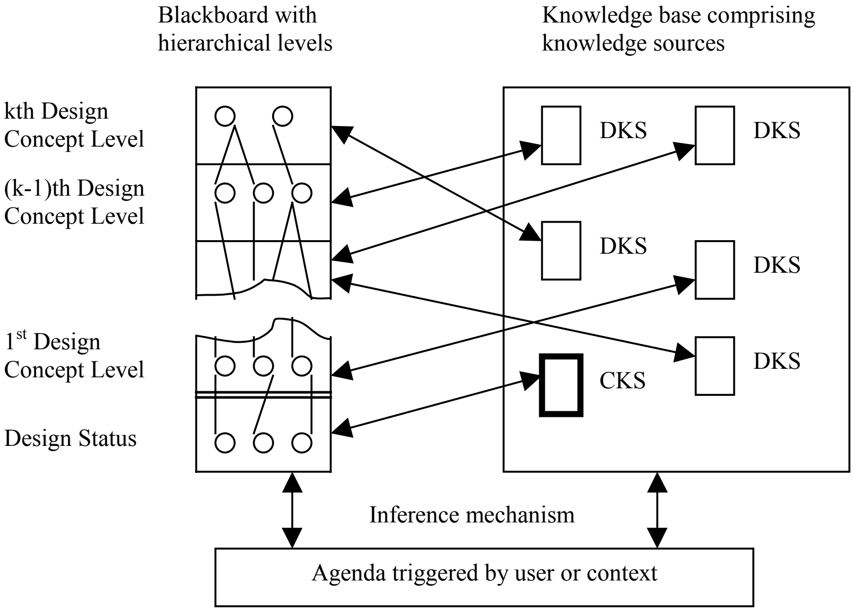

Blackboard with hierarchical levels
Knowledge base comprising knowledge sources
kth Design Concept Level
(k-1)th Design Concept Level
1st Design Concept Level
Design Status
DKS
DKS
DKS
DKS
DKS
CKS
Inference mechanism
Agenda triggered by user or context

Fig. 1. Blackboard architecture of the prototype expert system (DKS and CKS denote Domain Knowledge Source and Control Knowledge Source respectively)

3.1 Blackboard Architecture

Blackboard architecture has been developed to furnish a problem-solving model with contribution from a multitude of knowledge sources at different levels by integration into a single system. A variety of specialized expertise or knowledge sources are grouped into separate modules by employing both rules and frames and, sometimes under object-oriented programming environment. Blackboard system encapsulates information sharing through the common data structure called a blackboard, which compiles the data entries as well as acts as the communication link among various knowledge sources. The blackboard acts as the global system context, which stores the current state of the solution, including problem data, intermediate parameters and final outputs of the design.

Previously, the blackboard architecture has been applied in solving a diversity of problems in other areas: control [19], speech recognition [20], dynamic rescheduling [21], crankshaft design [22], damage assessment of steel bridge [23], control of a cryogenic cooling plant [24], large space structures [25], liquid retaining structure [26], etc. Figure 1 shows the blackboard architecture of the system, which consists of diverse knowledge sources, a blackboard and an inference mechanism. It is adopted since the reasoning with multiple knowledge sources is essential to solve the problem on design of storm drainage system, which usually entails interaction between diversified knowledge sources. Moreover, the design follows from opportunistic decisions, which are often made incrementally.

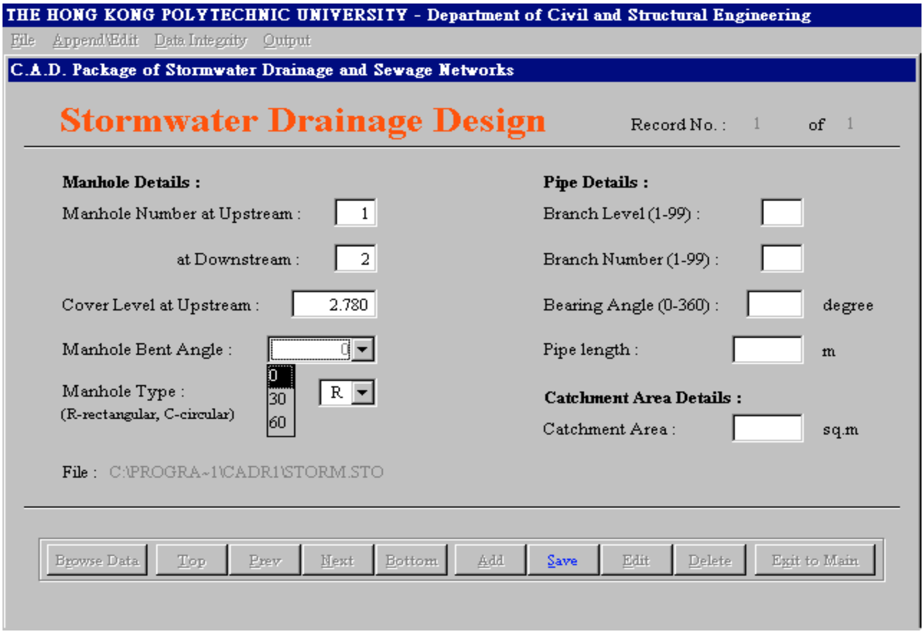
Under the declarative knowledge representation environment, objects are used to encapsulate knowledge structure, procedures, and values. The prototype system combines expert systems technologies, object-oriented programming, relational database models and graphics in Microsoft Windows environment. By defining various types of windows as different classes, such as Check Box, Option Button, List Box, Command Button, Text Box, etc., they can inherit common characteristics and possess their own special properties.

3.2 Knowledge Representation

The knowledge is represented in multi-formalism approach comprising object-oriented programming, rules, procedural methods, extensive numerical algorithm and databases in this system. Its knowledge base comprises representations of design knowledge of the drainage expert.

Design context and the processes in the design, both represented as objects, are organized separately. The objects define the static knowledge that represents design entities and their attributes, which can be either descriptive or procedural in form. The blackboard is partitioned into a number of hierarchical levels, corresponding to different stages of the design process. This kind of declarative knowledge is unable to effect program execution merely by itself, but the attribute values of different objects can be stored and retrieved whenever they are required during the problem solving process. This organization emulates closely the reasoning mechanism of a human expert designer. Either one of the following attribute types, namely, compound, multi-compound, instance reference, numeric, simple, string, interval, and time, is defined for each class. A facet designs the inference strategy for processing an attribute. A search order list is set optionally for each attribute, whose value is obtained from rules, session context, default value, method or end-user query.

Reasoning knowledge, both heuristic and judgmental, including the constraints between the objects, is represented as rules. Knowledge represented in the IF/THEN production rules with confidence factors can be assigned either automatically, or in response to the user’s request. These rules are a formal way of specifying how an expert drainage designer reviews a condition, considers various possibilities, and recommends an action. If the antecedent of a rule is determined to be true, the inference engine may fire the rule, inferring the conclusion statements to be true, which is then added to the working memory. Such rules are invoked mainly through change in pattern of other subprograms, instead of through a call from other subprograms in a specified algorithmic fashion. During each cycle, the conditions of each rule are matched against the current state of domain contexts. Rules are grouped into a rule set representing a collection of production rules with the same attribute as the conclusion. The rule sets include the knowledge necessary for the determination of different material properties, various geometrical ratios, interpolation of head loss coefficients, and selection of design parameters such as runoff coefficients in determination of peak runoff.


THE HONG KONG POLYTECHNIC UNIVERSITY - Department of Civil and Structural Engineering

C.A.D. Package of Stormwater Drainage and Sewage Networks

Stormwater Drainage Design

Record No. : 1 of 1

Manhole Details :

Manhole Number at Upstream : 1

at Downstream : 2

Cover Level at Upstream : 2.780

Manhole Bent Angle : 0

Manhole Type : R

(R-rectangular, C-circular)

Pipe Details :

Branch Level (1-99) :

Branch Number (1-99) :

Bearing Angle (0-360) : degree

Pipe length : m

Catchment Area Details :

Catchment Area : sq.m

File : C:\PROGRA~1\CADR1\STORM.STO

Fig. 2. Screen displaying interactive user interface of the system

Procedural knowledge, such as numerical processing, is represented in the form of object-oriented programming. Generic design entities are structured in a hierarchical knowledge base with inheritance properties. Besides, it comprises a blackboard together with two sets of knowledge sources, namely, Domain Knowledge Sources and Control Knowledge Sources, which represent the design processes. Diverse Domain Knowledge Sources, functioning independently and cooperatively through the blackboard, encode the actions to take for incremental build-up of the entire design process. Control Knowledge Sources involve meta-level knowledge, which establishes the problem solving strategy and controls the execution of the Domain Knowledge Sources. Since design steps in this system are explicitly seen on the main screen display, the sequence of design processes is primarily selected by the user. However, the validity of the sequencing is checked by Control Knowledge Sources.

Design Status only comprises a single object whereas there are several objects in the Design Concept level. Data inside Design Status are employed by the Control Knowledge Sources to determine the next possible action. After a specific design stage has been satisfied, the pertinent Design Status indicator will be assigned one of the values from the preset value list.

Most KBS development tools are not tailored for numerical processing but, instead, are designed for symbolic processing. However, this system can handle both symbolic and algorithmic programs simultaneously. Algorithmic models include capacity checking, velocity checking and backwater calculation. Custom-built codes as well as available existing codes are employed to perform these number-crunching tasks. Upon completion of execution of the external program, the previous session in the KBS is resumed. Of course, the Process Control knowledge modules continue to control the actions to be taken, depending on the outputs.

A database system is typically a record-keeping system employed to maintain relatively large amount of data. Some types of engineering knowledge are represented more conveniently in a database format. Here, database tables are used to represent engineering knowledge, such as head loss coefficients for various manhole configurations, fitting coefficients of the Intensity-Duration-Curves in Hong Kong, properties of pipe sections and properties of proposed alternative. Some heuristics are used to limit the choice of some design parameters to only practical values, acquired from practice engineers and code requirements. These databases contribute as a part of the entire design knowledge. Some of them such as the head loss coefficients are static and are not changed by any design activity whilst the others such as the database on properties of proposed alternative are dynamic and are generated during the execution of the system.

The knowledge used has been acquired mostly from written documents such as code of practice, textbooks and design manuals and complemented by experienced engineers involved with the design of storm drainage system. The user is guided through the process with step-by-step instructions. The system is capable of modeling and analyzing a network of up to 1,000 manholes. During execution, the screen shows the key parameters in one of the following forms: (i) flow number, pipe diameter, flow capacity, peak runoff; (ii) flow number, pipe diameter, actual flow velocity; (iii) flow number, pipe diameter, backwater level, and finished level, corresponding with the current calculation: i.e., capacity checking, velocity checking or backwater calculation.

3.3 Inference Methods

The inference engine controls the strategies on the selection of procedure methods and production rules and determines how, from where, and in what order a knowledge base draws its conclusions or design context. The Control Knowledge Sources evaluate the Design Status and decide the action in a data-driven forward chaining mechanism. The Domain Knowledge Sources need both forward and backward chaining inference mechanism to arrive at the solution. Based on the rated heuristic scores, Control Knowledge Sources then select the best action and proposes to the user before execution. All the design steps can be seen explicitly on the main screen display. This cycle is repeated until some feasible solutions satisfying all constraints are found.

The validity of the user's choice on the preferred sequence of design processes is checked by Control Knowledge Sources, which act opportunistically upon being triggered by user or situation during the design process. An event-driven inference processing mechanism is adopted so that the ensuing action of the system will depend on the input made by the user. For example, the applicable time of concentration is considered in accordance with the type of catchment. If developed area has been selected, the user is prompted to enter the time of entry and the pipe length. If natural area has been chosen, the user is prompted to enter the average fall from summit of catchment and the longest distance on the line of flow.

The input data entries by the user are kept at minimum, mostly through selection of appropriate values of parameters from the menus and answers to the queries made by the system. The input value will be rejected if it is not within the specified range. The system provides multi-window graphic images combined with valuable textual information, which is extremely valuable to novice designers. Figure 2 shows a typical screen displaying interactive user interface of the prototype system.

3.4 Application Examples

A typical storm drainage network in developed area with 39 manholes and a number of secondary and tertiary branches demonstrates the use of the prototype KBS. Each manhole should be numbered in a prescribed numbering order. Only the commercially available pipe diameters are given, i.e., multiple increments of 75 mm from 150 mm to 1,200 mm, and multiple increments of 150 mm from 1,200 mm to 2,100 mm. The design storm frequency, in accordance with the Hong Kong Civil Engineering Manual [18], is 1 in 50 year. The roughness value of all concrete pipes is assumed to be 0.6mm. For demonstration purpose of the capability of the KBS, arbitrary initial pipe diameters of 300 mm are input for all the pipes. The results have been verified rigorously with the conventional manual calculations, from which good agreements have been recorded. When performed manually, designing this system and producing design drawings required at least three iterations and 40 man-hours. Using the KBS, the same design did not require manual iteration and the design, including design drawings, was completed in only 1 man-hour. For larger storm drainage networks, the time savings become even greater.

3.5 Evaluation of System

In order to gauge the effectiveness of the system, 30 designers of varied technical backgrounds and experiences are required to complete a questionnaire with 8 questions that evaluate the presented system after their use. The feedback and written evaluations of the users on the scope and effectiveness of the system comprise several useful points. Table 1 shows the results of the user feedback questionnaire survey on using the system. Owing to the inherent variability in user rankings, only extreme rankings, such as exceeding a rank of '4-Agree', are considered significant. From the results, it is delighted to notice that no aspect of the system receives an unfavourable ranking. The tool is considered to be easy to comprehend, interesting, interactive, and relevant to designers. More importantly, the users find that it is extremely helpful and that the tool substantially increases their productivity in this application domain.

Table 1. Results of the user feedback questionnaire survey on using the system

Questionnaire item	Average rating#
Users basically possess good basic skill in using computer.	3.5
Users can actively control the design process through the system.	4.1
The system is easy to comprehend and follow.	4.2
The material with multiple formats of presentation is interesting.	4.0
The tool is interactive and user-friendly.	4.1
The presented material is relevant to the application domain.	4.2
The package is very helpful in assisting the design.	4.2
The tool substantially increases the productivity of users.	4.1

#1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree

4 Conclusions

It is demonstrated that the hybrid knowledge representation approach combining production rule system and object-oriented programming technique is viable with the implementation of blackboard system architecture for this domain problem. The knowledge base is transparent and is readily updated, which renders the KBS an ideal tool for incremental programming. By using custom-built interactive and user-friendly user interfaces, the prototype system is able to assist designers by furnishing with much needed expertise and cognitive support in the design activity. Some advantages of the KBS include improvement in efficiency and consistency of advice.

Acknowledgement. This research was supported by the Central Research Grant of Hong Kong Polytechnic University (G-T592) and the Internal Competitive Research Grant of Hong Kong Polytechnic University (A-PE26).

References

1. Chau, K.W.: An Expert System for the Design of Gravity-type Vertical Seawalls. Engineering Applications of Artificial Intelligence **5(4)** (1992) 363-367
2. Chau, K.W., Albermani, F.: Expert System Application on Preliminary Design of Liquid Retaining Structures. Expert Systems with Applications **22(2)** (2002) 169-178
3. Chau, K.W., Albermani, F.: Knowledge-Based System on Optimum Design of Liquid Retaining Structures with Genetic Algorithms. Journal of Structural Engineering ASCE **129(10)** (2003) 1312-1321
4. Chau, K.W., Albermani, F.: A Coupled Knowledge-Based Expert System for Design of Liquid Retaining Structures. Automation in Construction **12(5)** (2003) 589-602
5. Chau, K.W., Anson, M.: A Knowledge-Based System for Construction Site Level Facilities Layout. Lecture Notes in Artificial Intelligence **2358** (2002) 393-402
6. Chau, K.W., Chen, W.: An Example of Expert System on Numerical Modelling System in Coastal Processes. Advances in Engineering Software **32(9)** (2001) 695-703

7. Chau, K.W., Cheng, C., Li, C.W.: Knowledge Management System on Flow and Water Quality Modeling. Expert Systems with Applications **22(4)** (2002) 321-330
8. Chau, K.W., Ng, V.: A Knowledge-Based Expert System for Design of Thrust Blocks for Water Pipelines in Hong Kong. Water Supply Research and Technology - Aqua **45(2)** (1996) 96-99
9. Chau, K.W., Yang, W.W.: Development of an Integrated Expert System for Fluvial Hydrodynamics. Advances in Engineering Software **17(3)** (1993) 165-172
10. Chau, K.W., Yang, W.W.: A Knowledge-Based Expert System for Unsteady Open Channel Flow. Engineering Applications of Artificial Intelligence **5(5)** (1992) 425-430
11. Chau, K.W., Yang, W.W.: Structuring and Evaluation of VP-Expert Based Knowledge Bases. Engineering Applications of Artificial Intelligence **7(4)** (1994) 447-454
12. Chau, K.W., Zhang, X.Z.: An Expert System for Flow Routing in a River Network. Advances in Engineering Software **22(3)** (1995) 139-146
13. Chau, K.W.: A Robust Computer-Aided Design Package for Municipal Stormwater Drainage Networks. Advances in Engineering Software **15(1)** (1992) 43-53
14. Chau, K.W. and Ng, S.L.: A Robust Integrated Computer-Aided Design Package for Urban Drainage Networks. Water Science and Technology **30(1)** (1994) 117-120
15. Chau, K.W.: Integrated CAD Package for Storm-Water Drainage Networks. Journal of Water Resources Planning and Management, ASCE **121(4)** (1995) 336-339
16. Kitzmiller, C.T. Kowalik, J. S.: Coupling Symbolic and Numeric Computing in Knowledge-Based Systems. AI Magazine **Summer** (1987) 5-90
17. Ackers, P.: Charts for the Hydraulic Design of Channels and Pipes. CIRIA, London (1973)
18. Hong Kong Government: Civil Engineering Manual, Vol. IV: Sewerage and Drainage. Hong Kong Public Works Department, Hong Kong (1984)
19. Hayesroth, B.: A Blackboard Architecture for Control. Artificial Intelligence **26(3)** (1985) 251-321
20. Engelmore, R., Morgan, T.: Blackboard Systems. Addison_Wesley, Wokingham (1988)
21. Bharadwaj, A., Vinze, A.S., Sen, A.: Blackboard Architecture for Reactive Scheduling. Expert System with Applications **7(1)** (1994) 55-65
22. Lander, S.E., Corkill, D.D., Staley, S.M.: Designing Integrated Engineering Environments: Blackboard-Based Integration of Design and Analysis Tools. Concurrent Engineering Research and Applications **4(1)** (1996) 59-71
23. Barai, S.V., Pandey, P.C.: Integration of Damage Assessment Paradigms of Steel Bridges on a Blackboard Architecture. Expert System with Applications **19(3)** (2000) 193-207
24. Linkens, D.A., Abbod, M.F., Browne, A., Cade, N.: Intelligent Control of a Cryogenic Cooling Plant Based on Blackboard System Architecture. ISA Transactions **39(3)** (2000) 327-343
25. Kao, W.M., Adeli, H.: Multitasking Object-Oriented Blackboard Model for Design of Large Space Structures. Engineering Intelligent Systems for Electrical Engineering and Communications **10(1)** (2002) 3-8
26. Chau, K.W., Albermani, F.: Hybrid Knowledge Representation in a Blackboard KBS for Liquid Retaining Structure Design. Engineering Applications of Artificial Intelligence **17(1)** (2004) 11-18

Test Case Sequences in System Testing: Selection of Test Cases for a Chain (Sequence) of Function Clusters

Mark Sh. Levin and Mark Last

Ben-Gurion University of the Negev, Beer-Sheva, 84105, Israel
{marksh, mlast}@bgumail.bgu.ac.il

Abstract. The paper describes design of test case sequences in a multi-function system testing process. Groups of system functions (function clusters) are considered. It is assumed a set of test cases for each function cluster is designed or selected. The problem is: compose a test case sequence for a chain of function clusters (selection of a test case for each function cluster). Our approach is based on multicriteria decision making and Numerical examples illustrate the materials.

Keywords: Planning and Scheduling, System Design, System Testing

1 Introduction

In recent years, the significance of the analysis and testing phases for various complex systems is increasing ([2], [3], etc.). Contemporary systems are multidisciplinary ones and have a lot of different system functions which can be executed in various modes and various combinations. Here we consider black-box systems ([3], [8], etc.). Several related (e.g., by joint usage) system functions are considered as a function cluster and the preliminary **Problem 1** is [7]: *Find the best test case(s) for the function cluster.* In this article it is assumed the test cases for each function cluster are designed or selected ([4], [5], [7]). Thus we examine the next **Problem 2**: *Find the best test case sequence(s) for a chain of several function clusters.* This is a new problem that is formulated and solved here for the first time. Note the well-known traditional problem in system testing consists in the design of optimal (by length) test sequences via graph covering ([6], [9], etc.). Numerical examples illustrate the materials.

2 Testing Framework

Let $T = \{t_1, ..., t_i, ..., t_n\}$ be a set of test cases (a space). The following types of test cases and their groups are considered: *1.* elementary test case (test unit, as a set of input variables and a value for each variable): $t_i \in T$; *2.* group of independent test cases: $\{t_i\} \in T$; and *3.* sequence of test cases (multi-stage test case): $< t_{s[1]}, t_{s[2]}, ..., t_{s[k]} >$. The sequence corresponds to a chain-like use scenario. Now let us consider a set of system functions $F = \{f_1, ..., f_l, ..., f_L\}$. Each system function f_l corresponds to a set of system requirements [2]. Let $X(f_l)$ and $Y(f_l)$ be a set of inputs and outputs for function $f_l \in F$ accordingly. Evidently, it is necessary to examine properties of the system functions as follows: (i) structural parameters, e.g., corresponding system outputs

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 895–904, 2004.

and values; weights of "functional" importance; significance of possible corresponding fault(s));. (ii) binary relations on functions (independence, dependence, inclusion, interconnection by joint usage, etc.). Each cluster consists of several functions (medium hierarchical level) and corresponding *input-output* relationships (bottom hierarchical level). Note the function clusters can have intersections at the levels of *input-output* relationships and at the level of functions.

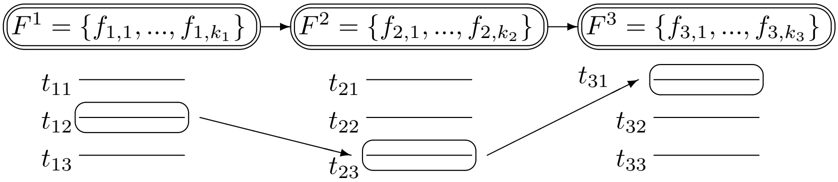

$F^1 = \{f_{1,1}, ..., f_{1,k_1}\}$
$F^2 = \{f_{2,1}, ..., f_{2,k_2}\}$
$F^3 = \{f_{3,1}, ..., f_{3,k_3}\}$
t_{11}
t_{12}
t_{13}
t_{21}
t_{22}
t_{23}
t_{31}
t_{32}
t_{33}

Fig. 1. Test case sequence for chain of function clusters

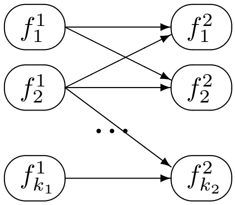


Fig. 2. Changes by functions

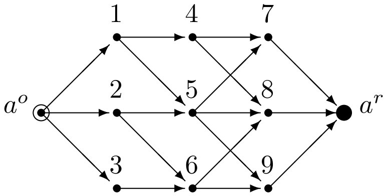


Fig. 6. Illustration for algorithm

Fig. 1 illustrates the design of test case sequence for three system function clusters $1, 2, 3$ with corresponding function sets F^1, F^2, F^3. For each function cluster a set of the most important test cases are generated (three for each function cluster in Fig. 1). The target series of test cases is depicted as follows: $< t_{12}, t_{23}, t_{31} >$. The solving scheme is the following:

Stage 1. Computation of basic (ideal) changes for the neighbor function clusters (as a vector with components corresponding to each input variable): $F^1 \rightarrow F^2$, $F^2 \rightarrow F^3$.

Stage 2. Computation for each pair of neighbor test cases (i.e., which correspond to neighbor function clusters) proximity to the ideal changes. This will be considered as a distance between the above-mentioned pair of the neighbor test cases.

Situation 2.1. Composition of the target series of test cases while taking into account the quality of the test cases (for the corresponding function clusters) and the quality of changes (as the above-mentioned distance). The problem is stated as a morphological clique problem (like for composition of test cases for the function cluster).

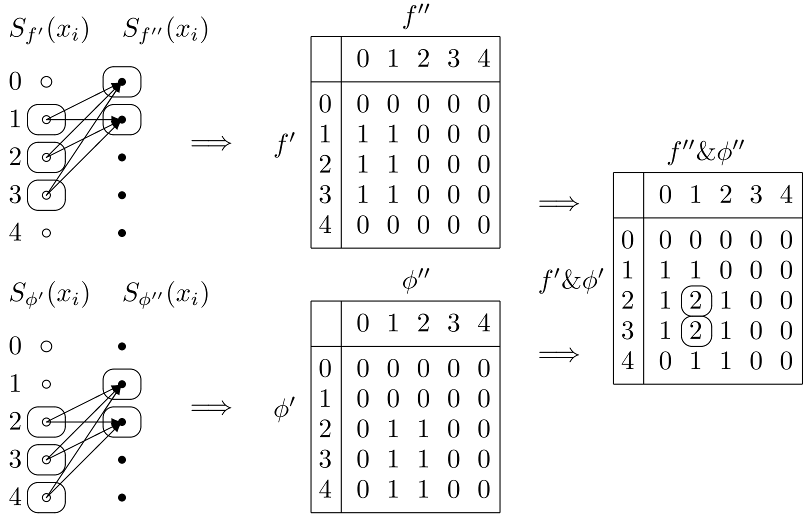
Situation 2.2. If the quality of the test cases for each function cluster is the same ones, it is possible to consider more simple problem: searching for the shortest path.

3 Interface between Neighbor Function Clusters

First, let us consider changes by functions. Fig. 2 depicts a bipartite digraph of the changes by functions. In general case (i.e., we have no information on transition by functions), the digraph is complete one. Further, we should like to define changes by *inputs* (on the basis of a numerical example): (a) changes for one variable $x_i \in X$ and several functions and building the ideal change (by values for the variable); (b) design of the corresponding ideal change as a vector of ideal value changes; (c) definition of the distance between the ideal change and a change by two individual functions (for ordinal and nominal scales).

Now we consider changes by *input* on the basis of four functions as follows: $f', \phi' \in F^1$ and $f'', \phi'' \in F^2$, Let input x_i is used by the functions: $x_i \in X(f')$, $x_i \in X(f'')$, $x_i \in X(\phi')$, and $x_i \in X(\phi'')$. We analyze two cases for x_i: (i) ordinal scale, and (ii) nominal scale. Let the ordinal scale for x_i be $S = [0,1,2,3,4]$ and $S_{f'}, S_{f''}, S_{\phi'}, S_{\phi''} \in S$. Fig. 3 depicts an example. We get the following two most important changes for x_i: (a) $2 \Rightarrow 1$ and (b) $3 \Rightarrow 1$. Thus a vector of the most important (*reference*) changes for all *inputs* $(x_1, ..., x_i, ..., x_n)$ is obtained (the changes can have alternatives as in the considered example). We define the *reference vector(s)* for two neighbor function clusters F^e and F^{e+1} as follows:

$$c^o(F^e, F^{e+1}) = (c^o_1(F^e, F^{e+1}), ..., c^o_i(F^e, F^{e+1}), ..., c^o_n(F^e, F^{e+1})).$$



f'', f':

	0	1	2	3	4
0	0	0	0	0	0
1	1	1	0	0	0
2	1	1	0	0	0
3	1	1	0	0	0
4	0	0	0	0	0

ϕ'', ϕ':

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	1	1	0	0
3	0	1	1	0	0
4	0	1	1	0	0

$f''\&\phi''$, $f'\&\phi'$:

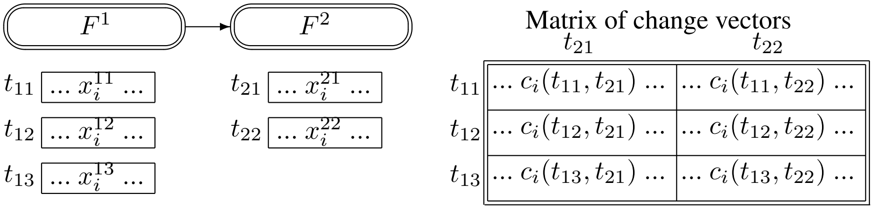
	0	1	2	3	4
0	0	0	0	0	0
1	1	1	0	0	0
2	1	$\boxed{2}$	1	0	0
3	1	$\boxed{2}$	1	0	0
4	0	1	1	0	0

Fig. 3. Changes by variable/values

Fig. 4 depicts two neighbor function clusters F^1, F^2 and corresponding composite test cases t_{11}, t_{12}, t_{13} (for the 1st cluster) and t_{21}, t_{22} (for the 2nd cluster). Comparison of the test cases lead to a matrix of change vectors by *inputs* (Fig. 4, right part).

Thus we get the change vector as follows: $c(t_{1u}, t_{2v}) = (\ ... \ , c_i(t_{1u}, t_{2v}), \ ... \ )$, where $1 \leq u \leq k_1$ (index by test cases for cluster F^1), $1 \leq v \leq k_2$ (index by test cases for cluster F^2). Let $c^o(F^1, F^2) = (\ ... \ , c^o_i(F^1, F^2), \ ... \ )$ be the *references vector(s)* and proximity between the examined vectors is the following:

$$\rho(c(t_{1u},t_{2v}),c^o(F^1,F^2)) = (\, ... \, , \rho_i(c_i(t_{1u},t_{2v}),c_i^o(F^1,F^2)), \, ... \,).$$


F^1
F^2
t_{11} $... x_i^{11} ...$
t_{12} $... x_i^{12} ...$
t_{13} $... x_i^{13} ...$
t_{21} $... x_i^{21} ...$
t_{22} $... x_i^{22} ...$

Matrix of change vectors

	t_{21}	t_{22}
t_{11}	$... c_i(t_{11},t_{21}) ...$	$... c_i(t_{11},t_{22}) ...$
t_{12}	$... c_i(t_{12},t_{21}) ...$	$... c_i(t_{12},t_{22}) ...$
t_{13}	$... c_i(t_{13},t_{21}) ...$	$... c_i(t_{13},t_{22}) ...$

Fig. 4. Change vectors

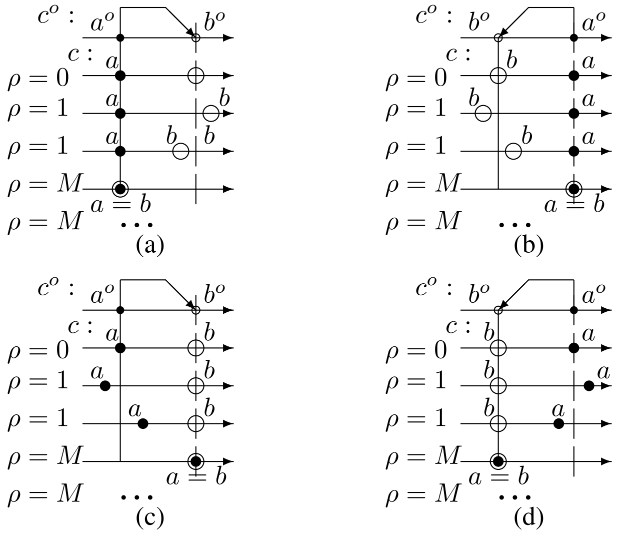


Fig. 5. Definition of proximity

Now we have to define the component of proximity $\rho_i(c_i(t_{1u},t_{2v}),c_i^o(F^1,F^2))$ for two kinds of scales: (i) ordinal scale and (ii) nominal scale. Let a^o and b^o be the initial and resultant values of $c_i^o(F^1,F^2)$ and a and b be the initial and resultant values of $c_i(t_{1u},t_{2v})$ accordantly. For their comparison, we have to examine the directions and points: (a) the same directions of the change (i.e., from a to b); (b) the inverse directions; (c) a direction and point (in the case $a^o = b^o$ or $a = b$); and (d) two points ($a^o = b^o$ and $a = b$). Fig. 6 depicts several situations and the resultant value of proximity (0 corresponds to equality). Thus our version of the definition for the proximity is the following:

1. Ordinal scale (and quantitative scale): *1.1* the case $a^o = b^o$: (i) $\rho = 0$ if $a = b = a^o$ and (ii) $\rho = M$ *elsewhere*, where M is a large number; *1.2* the case at Fig. 6a; *1.3* the case at Fig. 6b; *1.4* the case at Fig. 6c; *1.5* the case at Fig. 6d.

2. Nominal scale: (i) $\rho = 0$ if $a = a^o$ and $b = b^o$, (ii) $\rho = M$ *elsewhere.*

4 Dijkstra's Algorithm

Let $G = (A, E)$ be a digraph (the set of vertices $A$and the set of arcs E) with non-negative weights of arcs $(\alpha(e),\ e \in E)$. Here we briefly describe Dijkstra's algorithm (a discrete version of dynamic programming) to find the shortest path between two vertices (start vertex a^o and resultant vertex a^r) in digraph G. The algorithm is the following: *Stage 0.* Define the neighbor vertices for a^o: $A_1 \subseteq A$. Set for each corresponding two-vertex path $< a^o, a >,\ a \in A_1$ its weight as the weight of the arc. *Stage k.* Repeat the stage 0 $\forall a \in A_k,\ k = 2, 3, ...$ to define new vertices (set $A_k,\ k = 2, 3, ...$). Compute $\forall a \in A_k$ the best path from a^o (i.e., with the smallest total weight). *End stage.* Stop after the analysis of all vertices and select the smallest path to a^r. Evidently, each vertex is examined the only one time. Fig. 6 illustrates the algorithm: weights of arcs are the following: $\alpha(a^o, 1) = 1,\ \alpha(a^o, 2) = 2,\ \alpha(a^o, 3) = 3,\ \alpha(1, 4) = 4,\ \alpha(1, 5) = 5,\ \alpha(2, 5) = 6,\ \alpha(2, 6) = 1,\ \alpha(3, 6) = 2,\ \alpha(4, 7) = 2,\ \alpha(4, 8) = 5,\ \alpha(5, 7) = 3,\ \alpha(5, 8) = 2,\ \alpha(5, 9) = 4,\ \alpha(6, 8) = 1,\ \alpha(6, 9) = 4,\ (7, a^r) = 2,\ (8, a^r) = 2,\ (9, a^r) = 2$.

The paths after the *stage 0* are (weights are pointed out in brackets):
$w^0_1 =< a^o, 1 >\ (1),\ w^0_2 =< a^o, 2 >\ (2),\ w^0_3 =< a^o, 3 >\ (3)$.
The paths after the *stage 1* are (weights are pointed out in brackets):
$w^1_1 =< a^o, 1, 4 >\ (5),\ w^1_2 =< a^o, 1, 5 >\ (6),\ w^1_3 =< a^o, 2, 5 >\ (3)$.
The paths after the *stage 2* are (weights are pointed out in brackets):
$w^2_1 =< a^o, 1, 4, 7 >\ (7),\ w^2_2 =< a^o, 2, 5, 8 >\ (4),\ w^2_3 =< a^o, 2, 5, 9 >\ (7)$.

The resultant path is: $w^2_3 =< a^o, 2, 5, 8, a^r >\ (7)$. Here we considered a simple case (a layered digraph) that corresponds to our situation. In the case of vector-weights of the arcs (i.e., α is a vector with ordinal and / or quantitative elements), a set of Pareto-effective paths is selected at each stage $\forall a \in A_k$ and the result corresponds to the set of Pareto-effective paths too.

5 Design of Test Case Sequence

5.1 Analysis of Functions and Solving Scheme

In our example, a system structure is defined as follows: (a) 8 inputs (Table 1); (b) 5 outputs (Table 2); (c) 5 system functions: $f_1 : \{y_1, y_2\},\ f_2 : \{y_2, y_3, y_4\}$ (compressed input for y_4), $f_3 : \{y_1, y_2, y_3, y_4\}$ (compressed input for y_1, y_2, y_3, y_4, i.e., only x_3, x_4, x_6), $f_4 : \{y_4\}$, and $f_5 : \{y_5\}$. Interconnection *Input-Output* by system functions is presented in Table 3.

Here the structural relations are the following: (a) *independence*: $R^1_{x,y} : (f_1, f_5), (f_3, f_5), (f_4, f_5)$; $R^1_y : (f_2, f_5)$; (b) *dependency* or *intersection*: $R^2_{x_6} : (f_2, f_5),\ R^2_{y_2} : (f_1, f_2),\ R^2_{y_2,y_3,y_4} : (f_2, f_3)$, and $R^2_{y_1,y_2} : (f_1, f_3)$; and (c) *inclusion*: $R^3_y : (f_4 \subset f_3)$.

We examine a chain consisting of three function clusters as follows: $F^1 = \{f_1, f_3\}$ $F^2 = \{f_2, f_5\}$, and $F^3 = \{f_1, f_2, f_4\}$. Our solving process is the following: *1.* design of the composite test unit(s) for each function cluster (combinatorial synthesis); *2.* computing vectors for reference changes between F^1 and F^2, F^2 and F^3; *3.* computing proximity for real changes F^1 and F^2, F^2 and F^3 to the reference changes vectors; and *4.* design of the multi-stage (3-stage) composite test cases as the shortest path (Dijkstra's algorithm).

Table 1. Input variables

	Data type	Total # of values	Scale
x_1	Ordinal	3	$[0,1,2]$
x_2	Nominal	2	$[0,1]$
x_3	Nominal	2	$[0,1]$
x_4	Ordinal	5	$[0,1,2,3,4]$
x_5	Ordinal	3	$[0,1,2]$
x_6	Nominal	2	$[0,1]$

Table 2. Output variables

	Data type	Total # of values	Scale
y_1	Ordinal	2	$[0,1]$
y_2	Ordinal	5	$[0,1,2,3,4]$
y_3	Nominal	7	$[0,1,2,3,4,5,6]$
y_4	Ordinal	4	$[0,1,2,3]$
y_5	Nominal	4	$[0,1,2,3]$

Table 3. Interconnection *input-output* by functions

Input	Output				
	y_1	y_2	y_3	y_4	y_5
x_1	f_1, f_3	f_1, f_2, f_3	f_2, f_3	f_2, f_3, f_4	
x_2	f_1, f_3	f_1, f_2, f_3	f_2, f_3	f_3, f_4	
x_3	f_1	f_1, f_2	f_2		
x_4	f_1, f_3	f_1, f_2, f_3	f_2, f_3	f_2, f_3, f_4	
x_5					f_5
x_6			f_2		f_5

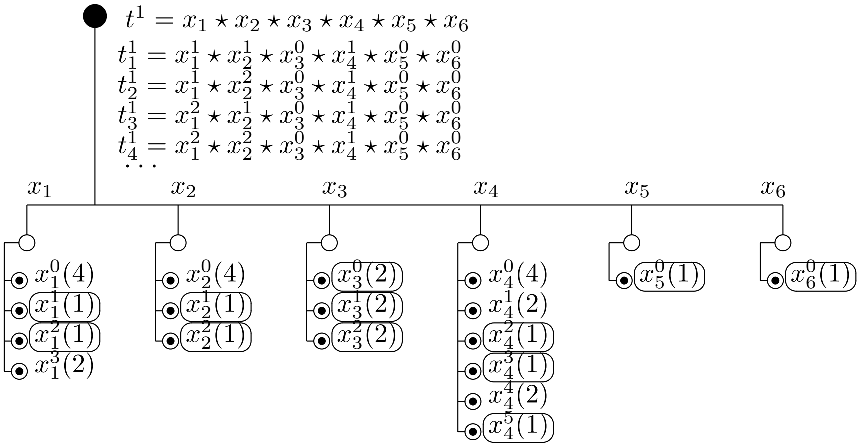

$t^1 = x_1 \star x_2 \star x_3 \star x_4 \star x_5 \star x_6$
$t^1_1 = x^1_1 \star x^1_2 \star x^0_3 \star x^1_4 \star x^0_5 \star x^0_6$
$t^1_2 = x^1_1 \star x^2_2 \star x^0_3 \star x^1_4 \star x^0_5 \star x^0_6$
$t^1_3 = x^2_1 \star x^1_2 \star x^0_3 \star x^1_4 \star x^0_5 \star x^0_6$
$t^1_4 = x^2_1 \star x^2_2 \star x^0_3 \star x^1_4 \star x^0_5 \star x^0_6$
...
x_1
x_2
x_3
x_4
x_5
x_6
$x^0_1(4)$
$x^1_1(1)$
$x^2_1(1)$
$x^3_1(2)$
$x^0_2(4)$
$x^1_2(1)$
$x^2_2(1)$
$x^0_3(2)$
$x^1_3(2)$
$x^2_3(2)$
$x^0_4(4)$
$x^1_4(2)$
$x^2_4(1)$
$x^3_4(1)$
$x^4_4(2)$
$x^5_4(1)$
$x^0_5(1)$
$x^0_6(1)$

Fig. 7. Structure of test cases for $\{f_1, f_3\}$

5.2 Test Cases for Function Clusters

We consider the synthesis of the most important test cases as a combination for input variables. Fig. 7 depicts composition of test cases for F^1 (on the basis of morphological clique problem [5]). Here 36 test cases are obtained. Analogically, solution groups are obtained for F^2 (36 test cases) and F^3 (16 test cases).

5.3 Simplification of Example

In previous section, we get a set of test cases for each function cluster. Now it is reasonable to demonstrate our approach on the basis of a simplified situation as follows (Fig. 8): (1) only part (3) of obtained test cases for F^1 will be considered: t^1_7, t^1_{20}, and t^1_{33}; (2) function cluster $F^2 = \{f_2, f_5\}$ will be change into $\overline{F^2} = \{f_2\}$ and corresponding test cases will be considered: t^2_1, t^2_{14}, and t^2_{23}; (3) two test cases will be considered for F^3: t^3_9 and t^3_{16}.

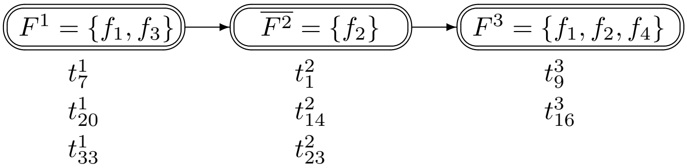


Fig. 8. Simplified situation

f_2

		x^0	0	1	2
	x^0	0	0	0	0
f_3	0	0	1	1	0
	1	0	1	1	0
	2	0	1	1	0

f_2

		x^0	0	1	2
	x^0	0	0	0	0
f_3	0	0	1	1	0
	1	0	1	1	0
	2	0	0	0	0

f_2

		x^0	0	1	2
	x^0	0	0	0	0
$f_1\&f_3$	0	0	②	②	0
	1	0	②	②	0
	2	0	1	1	0

Fig. 9. Changes for x_1 $(F^1 \Rightarrow \overline{F^2})$

Reference changes for x_2: f_2

		x^0	0	1
f_1	x^0	0	0	0
&	0	0	②	②
f_3	1	0	②	②

Reference changes for x_3: f_2

		x^0	0	1
f_1	x^0	0	①	①
&	0	0	1	①
f_3	1	0	①	①

Reference changes for x_4: f_2

		x^0	0	1	2	3	4
	x^0	0	0	0	0	0	0
	0	0	1	1	1	0	0
f_1	1	0	②	②	②	0	0
&	2	0	②	②	②	0	0
f_3	3	0	1	1	1	0	0
	4	0	1	1	1	0	0

Reference changes for x_6: f_2

		x^0	0	1
f_1	x^0	0	0	0
&	0	0	②	②
f_3	1	0	②	②

Fig. 10. Changes for $x_2, x_3, x_4, x_6,$ $(F^1 \Rightarrow \overline{F^2})$

5.4 Reference Changes, Proximity between Test Cases

The following basic (reference) changes are considered: *Cluster change 1.* $F^1 = \{f_1, f_3\} \Rightarrow \overline{F^2} = \{f_2\}$ and *Cluster change 2.* $\overline{F^2} = \{f_2\} \Rightarrow F^3 = \{f_1, f_2, f_4\}$. We examine variables (for both cases above): x_1, x_2, x_3, x_4, x_6. Fig. 9 presents changes for x_1 (*change 1*, x^0 corresponds to *None*). The process corresponds to Fig. 3. Other references changes for *change 1* (x_2, x_3, x_4, x_6) are presented in Fig. 10. Reference changes for *change 2* are presented in Figs. 11, 12.

Figs. 13, 14, 15, 16, and 17 present changes of test cases (c, the first line) and proximity (ρ, the second line). Definition of proximity (component x_i) is illustrated in Fig. 19: $\rho_i = \min_{l \in R_i} \{d_i(c_i, r_i^l)\}$, where $R_i = \{r_i^l\}$ is the set of reference changes for x_i and $d_i(c_i, r_i^l)$ is a distance from c_i to r_i^l.

Reference changes for x_1:

$f_1 \& f_2 \& f_4$

		x^0	0	1
f_2	x^0	0	0	0
	0	0	②	②
	1	0	②	②

Reference changes for x_2:

$f_1 \& f_2 \& f_4$

		x^0	0	1
f_2	x^0	0	0	0
	0	0	②	②
	1	0	②	②

Reference changes for x_3:

$f_1 \& f_2 \& f_4$

		x^0	0	1
f_2	x^0	0	①	①
	0	0	①	①
	1	0	①	①

Reference changes for x_4: $f_1 \& f_2 \& f_4$

		x^0	0	1	2	3	4
	x^0	0	0	0	0	0	0
	0	0	1	1	1	0	0
f_2	1	0	②	②	②	0	0
	2	0	②	②	②	0	0
	3	0	1	1	1	0	0
	4	0	1	1	1	0	0

Fig. 11. Changes for x_1, x_2, x_3, x_4, $(\overline{F^2} \Rightarrow F^3)$

Reference changes for x_5: $f_1 \& f_2 \& f_4$

		x^0	0	1	2
	x^0	0	0	0	0
f_2	0	0	③	③	1
	1	0	③	③	1
	2	0	0	0	0

Reference changes for x_6:

$f_1 \& f_2 \& f_4$

		x^0	0	1
f_2	x^0	0	0	0
	0	0	②	②
	1	0	②	②

Fig. 12. Changes for x_5, x_6 $(\overline{F^2} \Rightarrow F^3)$

	t_1^2
t_7^1	$(0 \to 1, 0 \to 0, 0 \to 0, 1 \to 0, x_5^0 \to x_5^0, x_6^0 \to 0)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ )$
t_{20}^1	$(1 \to 0, 1 \to 0, 0 \to 0, 2 \to 0, x_5^0 \to x_5^6, x_6^0 \to 0)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ )$
t_{33}^1	$(0 \to 0, 0 \to 0, 1 \to 0, 4 \to 0, x_5^0 \to x_5^0, x_6^0 \to 0)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0\ ,\ M\ )$

Fig. 13. Change c and proximity ρ $(F^1 \Rightarrow \overline{F^2})$

	t^2_{14}
t^1_7	$(1 \to 0, 0 \to 1, 0 \to 0, 1 \to 1, x^0_5 \to x^0_5, x^0_6 \to 0)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ )$
t^1_{20}	$(1 \to 0, 1 \to 1, 0 \to 0, 2 \to 1, x^0_5 \to x^0_5, x^0_6 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ )$
t^1_{33}	$(0 \to 0, 0 \to 1, 1 \to 0, 4 \to 1, x^0_5 \to x^0_5, x^0_6 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0\ ,\ M\ )$

Fig. 14. Change c and proximity ρ ($F^1 \Rightarrow \overline{F^2}$)

	t^2_{23}
t^1_7	$(1 \to 0, 0 \to 1, 0 \to 1, 0 \to 1, x^0_5 \to x^0_5, x^0_6 \to 1)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0\ ,\ M\ )$
t^1_{20}	$(1 \to 0, 1 \to 1, 0 \to 1, 2 \to 0, x^0_5 \to 0, x^0_6 \to 1)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ )$
t^1_{33}	$(0 \to 0, 0 \to 1, 1 \to 1, 4 \to 0, x^0_5 \to 0, x^0_6 \to 0)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0\ ,\ M\ )$

Fig. 15. Change c and proximity ρ ($F^1 \Rightarrow \overline{F^2}$)

	t^3_9
t^2_1	$(0 \to 1, 0 \to 0, 0 \to 0, 0 \to 1, x^0_5 \to x^0_5, 0 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ M\ ,\ M\ )$
t^2_{14}	$(0 \to 1, 1 \to 0, 0 \to 0, 1 \to 1, x^0_5 \to x^0_5, x^0_6 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ ,\ M\ )$
t^2_{23}	$(0 \to 1, 1 \to 0, 1 \to 0, 0 \to 1, 0 \to x^0_5, x^0_6 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ M\ ,\ M\ )$

Fig. 16. Change c and proximity ρ ($\overline{F^2} \Rightarrow F^3$)

	t^3_{16}
t^2_1	$(0 \to 1, 0 \to 1, 0 \to 1, 0 \to 2, x^0_5 \to x^0_5, 0 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ M\ ,\ M\ )$
t^2_{14}	$(0 \to 1, 1 \to 1, 0 \to 1, 1 \to 2, x^0_5 \to x^0_5, x^0_6 \to x^0_6)$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ M\ ,\ M\ )$
t^2_{23}	$(0 \to 1, 1 \to 1, 1 \to 1, 0 \to 2, 0 \to x^0_5, x^0_6 \to x^0_6$ $(\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ M\ ,\ M\ )$

Fig. 17. Change c and proximity ρ ($\overline{F^2} \Rightarrow F^3$)

5.5 Test Case Sequence

Fig. 19 depicts the design of the best 3-stage test case sequence (a version of Dijkstra's algorithm). We take into account proximities ρ from Figs. 13, 14, 15, 16, 17. The first stage of the algorithm leads to 2-vertex paths: (1) vertex t^2_1 (two Pareto-effective paths): $< t^1_7, t^2_1 >$ and $< t^1_{20}, t^2_1 >$; (2) vertex t^2_{14} (two Pareto-effective paths): $< t^1_7, t^2_{14} >$

and $< t_{20}^1, t_{14}^2 >$; (3) vertex t_{23}^2: $< t_{20}^1, t_{23}^2 >$. At the end stage we get the resultant 3-stage paths: $< t_7^1, t_{14}^2, t_9^3 >$, $< t_7^1, t_{14}^2, t_{16}^3 >$, $< t_{20}^1, t_{14}^2, t_9^3 >$, and $< t_{20}^1, t_{14}^2, t_{16}^3 >$.

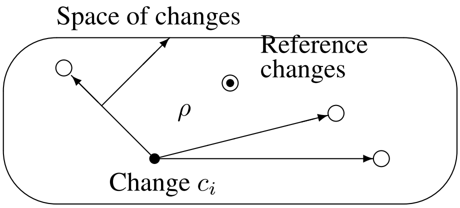


Fig. 18. Definition of proximity (x_i)

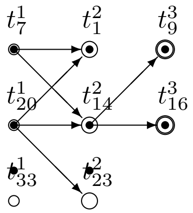


Fig. 19. 3-stage test cases

6 Conclusion

This paper presents our first step in design of test case sequences under multi-function system testing, which is a new problem i system-level testing of complex systems. Future research includes calculation of input changes for neighbor clusters based on function importance and usage frequency.

Acknowledgment. This work was partially supported by the National Inst. for Systems Test and Productivity at Univ. of South Florida under the USA Space and Naval Warfare Systems Command grant No. N00039-01-1-2248.

References

1. B. Chanrasekaran, J.R. Josephson, "Function in dence representation", *J. of Engineering with Computers*, **16** (3/4), 2000, 162-177.
2. D.M. Cohen, S.R. Dalal, J. Parelius, G.C. Patton, "The combinatorial design approach to automatic test generation," *IEEE Software*, Sept. 1996, 83-87.
3. P.C. Jorgensen, *Software Testing. A Craftsman's Approach.* CRC Press, Boca Raton, FL, 2002.
4. M. Last and A. Kandel, "Automated test reduction using an Info-Fuzzy Network", *Annals of Software Engineering*, 2003, 235–258.
5. M.Sh. Levin, *Combinatorial Engineering of Decomposable Systems*, Kluwer, Dordrecht, 1998.
6. M.Sh. Levin and M. Last, "Collection of test case sequences: Covering of function cluster digraph", *IASTED Conf. on AI and Applications*, Febr. 2004.
7. M.Sh. Levin and M. Last, "Multi-function system testing: Composition of Test Sets", *The 8th Int. Symp. HASE 2004*, March 2004.
8. P.J. Schroeder, B. Korel, "Black-box test reduction using Input-Output analysis", *AVM SIGSOFT Software Eng. Notes*, **25** (5), 2000, 173–177.
9. H. Ural, K. Zhu, "Optimal length test sequence generation using distinguishing sequences", *IEEE Trans. on Networking*, **1** (3), 1993, 358-371.

Supporting Constraint-Aided Conceptual Design from First Principles in Autodesk Inventor*

Alan Holland, Barry O'Callaghan, and Barry O'Sullivan

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
{a.holland|b.ocallaghan|b.osullivan}@cs.ucc.ie

Abstract. Engineering conceptual design can be defined as that phase of the product development process during which the designer takes a specification for a product to be designed and generates many broad solutions for it. It is well recognized that few computational tools exist that are capable of supporting the designer work through the conceptual phase of design. This paper presents a prototype constraint-based computer-aided design (CAD) technology, developed as an add-in to Autodesk Inventor, which can be used to support designers working in the early stages of design. The prototype has, at its core, a constraint filtering system based on generalized arc-consistency processing and backtrack search. We present aspects of our current prototype, focusing in particular on those aspects related to the interactive specification, development and configuration of the designer's concepts from an initial high-level specification.

1 Introduction

Engineering conceptual design can be regarded as that phase of the engineering design process during which the designer takes a specification for a product to be designed and generates many broad solutions for it. Each of these broad solutions is generally referred to as a scheme [7]; however, in this paper we will interchangeably use the terms scheme, design and concept. It is generally accepted that conceptual design is one of the most critical phases of the product development process. It has been reported that more than 75% of a product's total cost is dictated by decisions made during the conceptual phase of design and that poor conceptual design can never be compensated for at the detailed design stage [10].

The technology presented here has many features that are crucial for supporting designers during the conceptual phase of design. For example, designers can specify an initial statement describing the desired properties of the required artifact. The designer is free to modify this at any time by adding/removing constraints. Furthermore, the

* This research is funded by Enterprise Ireland, through their Research Innovation Fund (Grant Number RIF-2001-317). The software used for the project, Autodesk Inventor, has been sponsored by *cadcoevolution.com*, an Irish CAD tool provider.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 905-914, 2004.

designer has the freedom to approach the design process in any way he wishes. The prototype ensures consistency amongst the designer's decisions and can explain why inconsistencies have occurred. Finally, designers prefer to use tools which are familiar to them and, therefore, any additional tools that a designer is expected to use must have a "look-and-feel" similar to those they already use. It was these considerations that set the agenda for the work reported here.

This paper presents a prototype constraint-based computer-aided design (CAD) technology that can be used to support designers working in the early stages of design. The CAD technology has, at its core, a constraint filtering system based on generalized arc-consistency processing [3] and backtrack search.

The remainder of the paper is organized as follows. Section 2 presents an overview of the relevant literature. Section 3 presents an overview of the theory of conceptual design upon which the prototype is based. Section 4 presents a summary of some of the important features of the CAD system. Section 5 presents a walk-through the current prototype, describing in some detail how a scheme can be developed and configured by the designer. In Section 6 a number of concluding remarks are made.

2 Background

In the design literature three phases of design are generally identified: conceptual design, embodiment design and detailed design [16]. Constraint-based applications for design have been more commonly applied to the post-conceptual phases of design [11, 12, 21]. The use of constraint processing techniques for supporting configuration design has also been widely reported in the literature [13, 19]. Applications of constraints to supporting Concurrent Engineering, Integrated Product Development and agent-based design have also received significant interest [2, 4, 5, 9, 11, 17].

Constraint-based approaches to supporting conceptual design have been reported in the literature for quite a number of years [8, 18, 20]. However, most of this research does not address the synthesis problem; the vast majority has focused on constraint propagation and consistency management relating to more numerical design decisions. For example, "Concept Modeler" is based on a set of graph processing algorithms that use bipartite-matching and strong component identification for solving systems of equations [20]. The Concept Modeler system allows the designer to construct models of a product using iconic abstractions of machine elements. Based on the earlier work on Concept Modeler, a system called "Design Sheet" has been developed [18]. This system is essentially an environment for facilitating flexible trade-off studies during conceptual design. It integrates constraint management techniques, symbolic mathematics and robust equation solving capabilities with a flexible environment for developing models and specifying tradeoff studies.

While not a constraint-based system, the Conceptual Understanding and Prototyping Environment (CUP) is an approach to supporting conceptual design that unites ideas from traditional mechanical design with 3D layouts and knowledge engineering [1]. However, our technology is entirely constraint-based which gives us the opportunity to exploit the semantics of constraints and use inference as a core technique for navigating the design search space and providing explanations, as well as a declarative approach to modeling the evolving schemes that the designer wishes to explore.

3 Underlying Engineering Design Principles

The model of conceptual design adopted here is based on the generally accepted observation that during this phase of design a designer works from an informal set of requirements that the product must satisfy and generates alternative schemes consistent with them. Central to the process of scheme generation is an understanding of function and how it can be provided. The process involves the development of a functional decomposition that provides the basis for a realization of physical elements that form a scheme. In addition to determining which physical elements comprise a scheme, the relations between them must also be specified to a sufficient extent to permit the evaluation and comparison of alternative schemes.

In the remainder of this section a brief overview of some of the most important aspects of our approach to conceptual design will be presented. For a more complete discussion of the theory the reader is encouraged to refer to the more detailed literature available [14, 15].

The Design Specification. The conceptual design process is initiated by the recognition of a need or customer requirement. This need is analyzed and translated into a statement that defines the *functionality* that the product should provide and the *physical* requirements that the product must satisfy.

Conceptual Design Knowledge. We employ a *function-means map* approach to cataloging how function can be provided by means [6, 14]. In a function-means map two different types of means can be identified: *design principles* and *design entities*. A design principle is a means that is defined in terms of a set of functions that must be provided in order to provide some higher-level functionality. Design principles are abstractions of known approaches to providing function. By utilizing a design principle the designer can decompose higher-level functions without committing to a physical solution too early in the design process. The functions that are required by a design principle collectively replace the function being embodied by that principle. The functions that define a design principle will, generally, have a number of *context relations* defined between them. These context relations describe how the parts in the scheme, which provide these functions, should be configured so that the design principle is used in a valid way. Note that a design principle is not just a model of a known physical design solution, but is an abstraction that can be used to encourage the designer to develop the design space to be explored. A design entity is defined by a set of parameters and the constraints that exist between these parameters. For example, an electronic resistor would be modeled as a design entity that is defined by three parameters, resistance, voltage and current, between which Ohm's Law (a constraint) would hold.

Scheme Configuration using Interfaces. Generally, the first means that a designer will select will be a design principle. This design principle will substitute the required (parent) functionality with a set of child functions. Ultimately the designer will embody all leaf-node functions in the functional decomposition with design entities. During this embodiment process, the context relations from the design principles used in the scheme will be used as a basis for defining the interfaces (constraints) between the design entities. The types of interfaces that may be used to synthesize a product will be specific to the engineering domain within which the designer is working. Indeed, these interfaces may also be specific to the particular company to which the designer belongs in order to ensure the configurability of the product.

4 Features of the Current Prototype

In this section we briefly present the key features of our prototype CAD system for supporting conceptual design, which we call ConCAD Expert. The technology is seamlessly integrated with Autodesk Inventor[1]. This particular CAD system has been chosen for a number of reasons. In particular, as well as being one of the most popular 3D solid-modeling design environments, Inventor has an architecture similar to most tools of its kind, but has a very rich API through which we can integrate with the host CAD system. Our technology has, at its core, an interactive constraint filtering system based on generalized arc-consistency [3] and backtrack search. The system is fully interactive, monitoring the consistency of the designer's decisions and providing feedback when an inconsistency has been detected or the designer has requested justifications or explanations from the system. It was developed using C#, and uses the Autodesk Inventor 5.3 COM API. An XML database has been developed to store the parts available to the CAD system. The XML schema that has been used represents a part file containing the various attributes of each part. The CAD system is capable of the following:

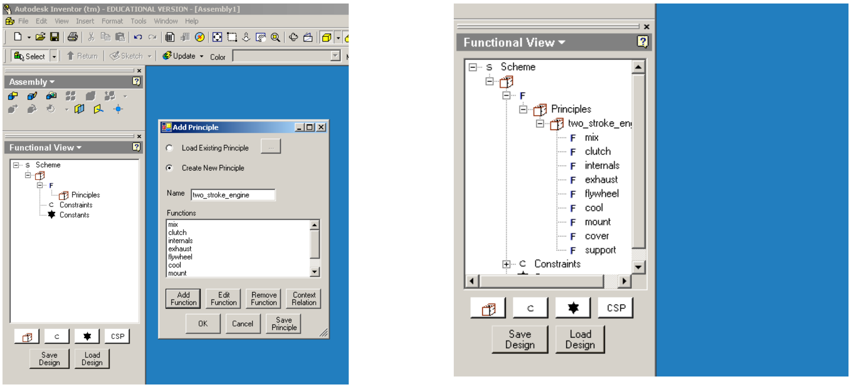
1. It can reason about both physical and functional design requirements. This has been enabled through the implementation of a constraint-based meta-layer within the host CAD system;
2. In terms of functional features, the CAD technology can use abstract descriptions of design principles to decompose the functional requirement of a design. Design principles can be created and stored on-the-fly for future use;
3. The technology supports inference (generalized consistency processing) over designer-specified/implied constraints. In addition, the configuration of parts and assemblies in the CAD system is checked for consistency against the constraints represented in the design specification, constraints introduced as a result of the design principles incorporated by the designer, and constraints arising from the definition of parts and subassemblies.
4. The decisions underpinning the functional and physical synthesis of the design are available at all times and can be regarded as a history of the design's evolution.
5. While the system ensures consistency amongst a designer's decisions, it does not strictly forbid any action that the designer may wish to take. In this way the technology acts as an intelligent assistant that provides advice on the consequences of decisions, rather than preventing the designer from making inconsistent ones.

[1] See `http://www.autodesk.com/inventor`

However, a number of issues are not addressed by the current prototype. Amongst these is support for freeform sketching. Also, the role of intelligent user-interfaces for conceptual design is outside the scope of our current work.

5 An Interactive Design Session

In this section, a detailed interaction between a designer and the CAD system is presented. Not all features of the system are presented. Rather, we focus here on how the functional decomposition of a design is developed, and how this impacts the physical design. In the example scenario we consider here, we simulate an interaction with a designer who wishes to develop a concept for a two-stroke engine. Our focus here is to simply present how our system supports the development of a concept from first principles.



○ Load Existing Principle

● Create New Principle

Name two_stroke_engine

Functions

mix
clutch
internals
exhaust
flywheel
cool
mount

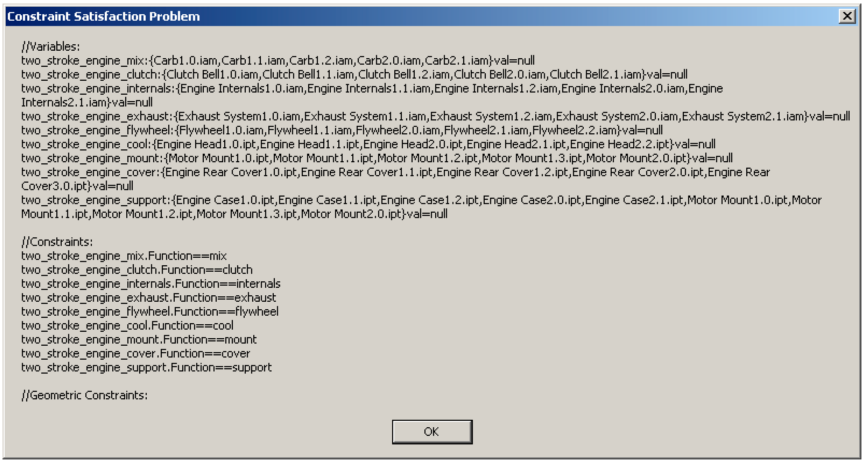
(a) (b)

Fig. 1. Beginning our design project: deciding on a principle we wish to employ.

In Figure 1, the designer has set about developing a concept for the desired artifact. In Figure 1(a), we see the starting state of the CAD system that primarily comprises a Functional View of the product. This panel is used by the designer to specify the functional decomposition of the product, the initial functional requirements, the physical requirements, and any additional constraints that are added during the design process.

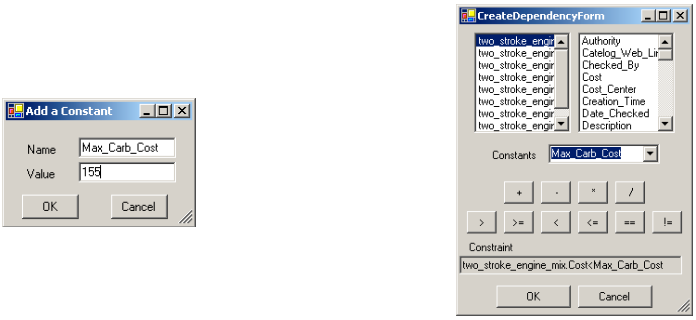
Also in Figure 1(a) the designer specifies a new design principle that he wishes to employ in the new design. This new design principle, based on a two-stroke engine, comprises a number of functions that the user specifies. Figure 1(b) illustrates the function decomposition after the designer incorporates the new principle into his design. Clearly, we can see that the designer now must consider embodying several functions in order to develop a concept based on the principle he has employed.

The consequences of the designer's decisions up to this point can be clearly seen in Figure 2. As the designer develops a concept, the CAD system builds a constraint satisfaction problem representing the design. It is based upon this model that the inference capabilities of the CAD system reason about the consistency of the designers decisions. In Figure 2 we can see that there are a set of variables, each with corresponding domains, and some functional constraints. We can also see that the designer has not yet entered any physical or geometrical constraints.



```
//Variables:
two_stroke_engine_mix:{Carb1.0.iam,Carb1.1.iam,Carb1.2.iam,Carb2.0.iam,Carb2.1.iam}val=null
two_stroke_engine_clutch:{Clutch Bell1.0.iam,Clutch Bell1.1.iam,Clutch Bell1.2.iam,Clutch Bell2.0.iam,Clutch Bell2.1.iam}val=null
two_stroke_engine_internals:{Engine Internals1.0.iam,Engine Internals1.1.iam,Engine Internals1.2.iam,Engine Internals2.0.iam,Engine
Internals2.1.iam}val=null
two_stroke_engine_exhaust:{Exhaust System1.0.iam,Exhaust System1.1.iam,Exhaust System1.2.iam,Exhaust System2.0.iam,Exhaust System2.1.iam}val=null
two_stroke_engine_flywheel:{Flywheel1.0.iam,Flywheel1.1.iam,Flywheel2.0.iam,Flywheel2.1.iam,Flywheel2.2.iam}val=null
two_stroke_engine_cool:{Engine Head1.0.ipt,Engine Head1.1.ipt,Engine Head2.0.ipt,Engine Head2.1.ipt,Engine Head2.2.ipt}val=null
two_stroke_engine_mount:{Motor Mount1.0.ipt,Motor Mount1.1.ipt,Motor Mount1.2.ipt,Motor Mount1.3.ipt,Motor Mount2.0.ipt}val=null
two_stroke_engine_cover:{Engine Rear Cover1.0.ipt,Engine Rear Cover1.1.ipt,Engine Rear Cover1.2.ipt,Engine Rear Cover2.0.ipt,Engine Rear
Cover3.0.ipt}val=null
two_stroke_engine_support:{Engine Case1.0.ipt,Engine Case1.1.ipt,Engine Case1.2.ipt,Engine Case2.0.ipt,Engine Case2.1.ipt,Motor Mount1.0.ipt,Motor
Mount1.1.ipt,Motor Mount1.2.ipt,Motor Mount1.3.ipt,Motor Mount2.0.ipt}val=null

//Constraints:
two_stroke_engine_mix.Function==mix
two_stroke_engine_clutch.Function==clutch
two_stroke_engine_internals.Function==internals
two_stroke_engine_exhaust.Function==exhaust
two_stroke_engine_flywheel.Function==flywheel
two_stroke_engine_cool.Function==cool
two_stroke_engine_mount.Function==mount
two_stroke_engine_cover.Function==cover
two_stroke_engine_support.Function==support

//Geometric Constraints:
```

Fig. 2. The constraint satisfaction problem (CSP) representation of the design.



Name Max_Carb_Cost
Value 155

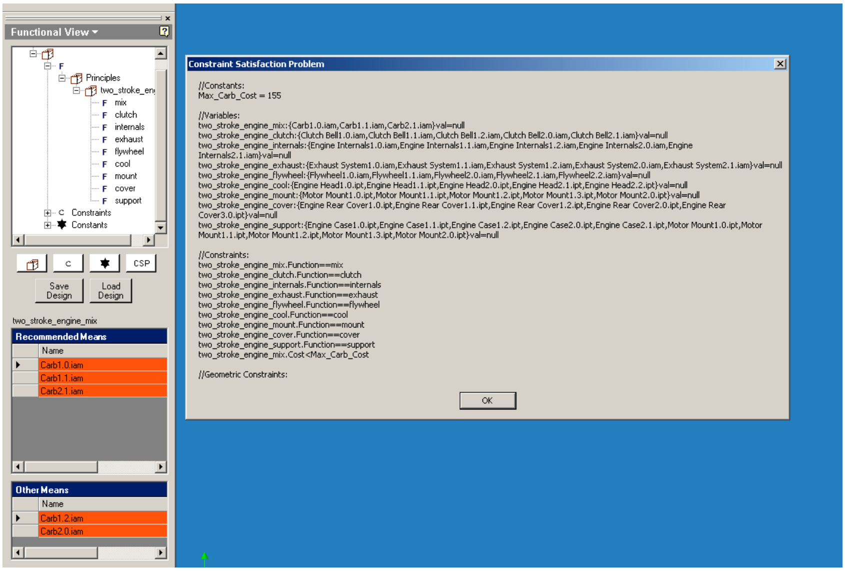
Constants Max_Carb_Cost
Constraint
two_stroke_engine_mix.Cost<Max_Carb_Cost

(a) (b)

Fig. 3. Adding additional constraints to the model

However, in Figure 3 the designer adds additional constraints into the model. In Figure 3(a) the designer defines a constant (a variable and a unary constraint) stating that the maximum cost of a carburetor element in the design must be 155 units. This constant is used in the definition of a more complex constraint in Figure 3(b) relating the constant with a property of the design. Specifically, the designer ensures that the maximum cost of the embodiment of the "mix" function is less than the maximum allowed value. In Figure 4 we see the consequences of this decision on the choices that the designer has available.

Figure 4 presents the interface of the CAD system after the designer has made several more decisions. Firstly, once the designer specified the principle underpinning the function decomposition, by clicking on particular leaves of the function tree, the CAD system can advise on consistent means for providing that function. Using a color-coding system, the designer can distinguish between sub-assemblies, parts and design principles. Furthermore, the means that are available to the designer are partitioned into two sets: the set of "Recommended Means" and the set of "Other Means". The former is the set of means that can provide the desired functionality and are also consistent with the physical constraints in the CSP model. On the other hand, the latter is the set of means that, while satisfying the functional constraints in the model, violate at least one physical constraint. Both sets are shown so that the designer can see the consequences of his decisions. Explanations and detailed property information is available by clicking on a means. The information provided will help the designer determine why a particular means is no longer recommended for use.



two_stroke_engine_mix

Recommended Means

Name
Carb1.0.iam
Carb1.1.iam
Carb2.1.iam

Other Means

Name
Carb1.2.iam
Carb2.0.iam

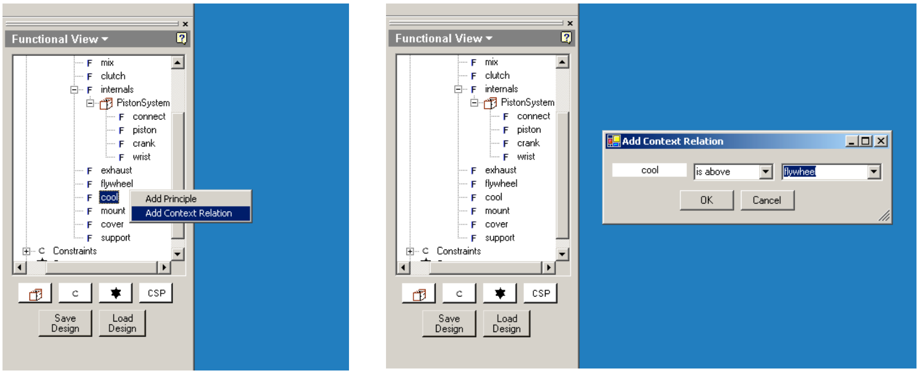
```
//Constants:
Max_Carb_Cost = 155

//Variables:
two_stroke_engine_mix:{Carb1.0.iam,Carb1.1.iam,Carb2.1.iam}val=null
two_stroke_engine_clutch:{Clutch Bell1.0.iam,Clutch Bell1.1.iam,Clutch Bell1.2.iam,Clutch Bell2.0.iam,Clutch Bell2.1.iam}val=null
two_stroke_engine_internals:{Engine Internals1.0.iam,Engine Internals1.1.iam,Engine Internals1.2.iam,Engine Internals2.0.iam,Engine
Internals2.1.iam}val=null
two_stroke_engine_exhaust:{Exhaust System1.0.iam,Exhaust System1.1.iam,Exhaust System1.2.iam,Exhaust System2.0.iam,Exhaust System2.1.iam}val=null
two_stroke_engine_flywheel:{Flywheel1.0.iam,Flywheel1.1.iam,Flywheel2.0.iam,Flywheel2.1.iam,Flywheel2.2.iam}val=null
two_stroke_engine_cool:{Engine Head1.0.ipt,Engine Head1.1.ipt,Engine Head2.0.ipt,Engine Head2.1.ipt,Engine Head2.2.ipt}val=null
two_stroke_engine_mount:{Motor Mount1.0.ipt,Motor Mount1.1.ipt,Motor Mount1.2.ipt,Motor Mount1.3.ipt,Motor Mount2.0.ipt}val=null
two_stroke_engine_cover:{Engine Rear Cover1.0.ipt,Engine Rear Cover1.1.ipt,Engine Rear Cover1.2.ipt,Engine Rear Cover2.0.ipt,Engine Rear
Cover3.0.ipt}val=null
two_stroke_engine_support:{Engine Case1.0.ipt,Engine Case1.1.ipt,Engine Case1.2.ipt,Engine Case2.0.ipt,Engine Case2.1.ipt,Motor Mount1.0.ipt,Motor
Mount1.1.ipt,Motor Mount1.2.ipt,Motor Mount1.3.ipt,Motor Mount2.0.ipt}val=null

//Constraints:
two_stroke_engine_mix.Function==mix
two_stroke_engine_clutch.Function==clutch
two_stroke_engine_internals.Function==internals
two_stroke_engine_exhaust.Function==exhaust
two_stroke_engine_flywheel.Function==flywheel
two_stroke_engine_cool.Function==cool
two_stroke_engine_mount.Function==mount
two_stroke_engine_cover.Function==cover
two_stroke_engine_support.Function==support
two_stroke_engine_mix.Cost<Max_Carb_Cost

//Geometric Constraints:
```

Fig. 4. Part of the CAD system interface after the designer has introduced a new constraint.

Also shown in Figure 4 is the current CSP model encoding the designer's decisions. It can be clearly seen that the set of means being recommended for the "mix" function has been reduced. The new constant definition and cost constraint can also be seen in the model.

Earlier, when describing how design principles could be specified, we did not illustrate a context relation being defined. Context relations are critical parts of the definition of a design principle in conceptual design since they specify how a design principle is to be used in a valid manner when parts are being introduced into the scheme.

Figure 5 presents a sequence of interactions that a designer can use to specify/modify the context relations associated with functions in a design principle. In Figure 5(a) the designer selects the "cool" function of the two-stroke engine principle. In Figure 5(b) the designer states what the relation between this function and another function should be. In this case the designer states that a spatial relationship must hold between the part(s) providing the cooling function in the engine and the part(s) providing the flywheel functionality. However, in general, context relations can also define how parts are interfaced with each other. We shall see below how this affects the placement of parts.

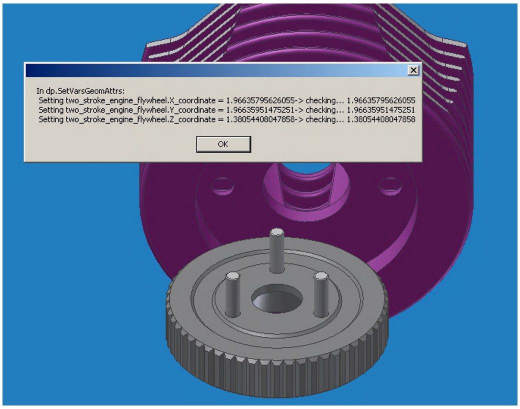


cool is above flywheel

(a) (b)

Fig. 5. Defining a context relation between functions in a design principle.

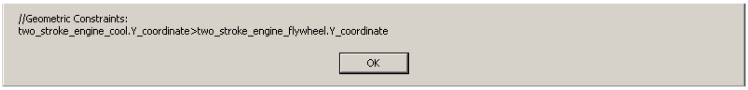
However, before moving on, it should be noted that the functional decomposition in this figure is more detailed than previously. This is due to the designer employing another design principle into the design. It can be clearly seen how the functional decomposition can be developed during design, becoming a complex tree of functions and relationships between them that must be satisfied when a collection of parts is used to provide the necessary functionalities.



```
In dp.SetVarsGeomAttrs:
 Setting two_stroke_engine_flywheel.X_coordinate = 1.96635795626055-> checking... 1.96635795626055
 Setting two_stroke_engine_flywheel.Y_coordinate = 1.96635951475251-> checking... 1.96635951475251
 Setting two_stroke_engine_flywheel.Z_coordinate = 1.38054408047858-> checking... 1.38054408047858
```

Fig. 6. Positioning the flywheel and the cooling element to satisfy the spatial context relation.

In Figure 6 we see how a context relation affects the placement of parts in the CAD system. Depicted here is a scenario in which the designer has selected a part to provide the cooling functionality and a second part for the flywheel functionality. As the designer positions the part for the latter, a geometric constraint, in this case a spatial constraint, is being checked for consistency. Figure 6 shows the result of this check, indicating, in this case, that the constraint has been satisfied.



//Geometric Constraints:
two_stroke_engine_cool.Y_coordinate>two_stroke_engine_flywheel.Y_coordinate

Fig. 7. Part of the CSP model, after the designer has incorporated parts for the cooling and flywheel functions, showing the geometric/spatial constraint implied by a context relation.

In Figure 7 we can see part of the current state of the CSP model as these interactions are taking place. Note the geometric constraint in the figure. It is this constraint that was being propagated and checked in Figure 6. As the designer develops the concept further, employing new design principles, incorporating parts into the design, etc., this CSP model will become far more detailed. Figure 8 presents an example state of the CAD system after the designer has made several more decisions.

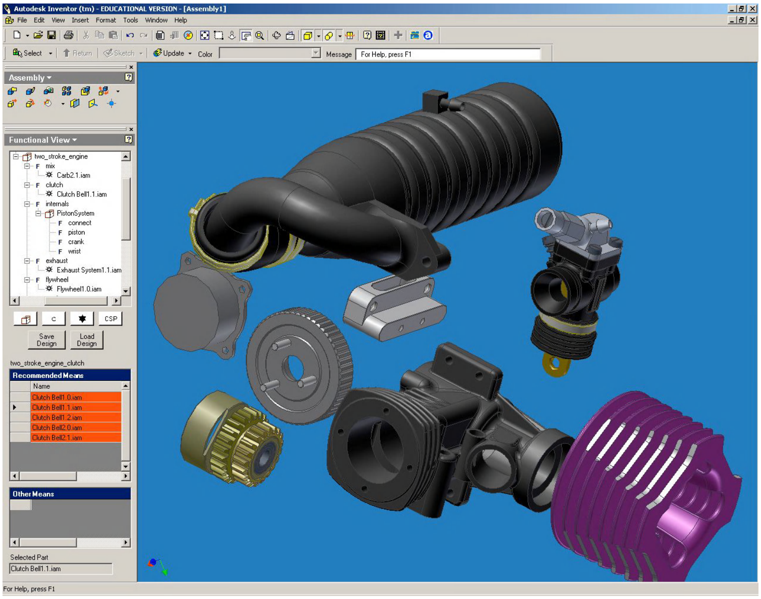


Fig. 8. The interface of the CAD system after many more interactions with the designer.

Note that in the process we have described the designer is free to make any decisions that he wishes at any stage in the design process. The CAD technology we describe simply maintains the consistency of the designer's decision, but does not strictly forbid any form of action. This is a critical characteristic that CAD technologies for supporting conceptual design must exhibit.

6 Conclusions

While engineering conceptual design is regarded as the most critical phases of product development few computational tools exist that are capable of supporting the designer work through the conceptual phase of design. This paper presented a prototype constraint-based computer-aided design (CAD) technology, developed as an add-in to Autodesk Inventor, which can be used to support designers working in the early stages of design. The add-in has, at its core, a constraint filtering system based on generalized arc-consistency processing and backtrack search. We presented aspects of our current prototype, focusing on those aspects related to the interactive specification, development and configuration of the designer's concepts from first principles.

References

1. L. Anthony, W.C. Regli, J.E. John, and S.V. Lombeyda. CUP: A computer-aided conceptual design environment for assembly modeling. Technical Report DU-MCS-01005, Department of Mathematics and Computer Science, Drexel University, June 2001.
2. D. Bahler, C. Dupont, and J. Bowen. An axiomatic approach that supports negotiated resolution of design conflicts in Concurrent Engineering. In Artificial Intelligence in Design, pages 363–379, 1994.
3. C. Bessiere and J.-C. Regin. Arc consistency for general constraint networks: preliminary results. In Proceedings IJCAI'97, pages 398–404, 1997.
4. W.P. Birmingham and A. Ward. What is Concurrent Engineering? AIEDAM, 9:67–68, 1995. Guest Editorial in a Special Issue on Concurrent Engineering.
5. J. Bowen and D. Bahler. Frames, quantification, perspectives and negotiation in constraint networks in life-cycle engineering. Artificial Intelligence in Engineering, 7:199–226, 1992.
6. J. Buur. A Theoretical Approach to Mechatronics Design. PhD thesis, TU Denmark, 1990.
7. M.J. French. Engineering Design: The Conceptual Stage. Heinemann, London, 1971.
8. M.D. Gross, S.M. Ervin, J.A. Anderson, and A. Fleisher. Constraints: Knowledge representation in design. Design Studies, 9(3):133–143, July 1988.
9. D. Haroud, S. Boulanger, E. Gelle, and I. Smith. Management of conflict for preliminary engineering design tasks. AIEDAM Journal, 9:313–323, 1995.
10. W. Hsu and B. Liu. Conceptual design: Issues and challenges. CAD, 32(14):849–850, 2000.
11. C. Lottaz, I.F.C. Smith, Y. Robert-Nicoud, and B.V. Faltings. Constraint-based support for negotiation in collaborative design. Artificial Intelligence in Engineering, 14:261–280, 2000.
12. C. Lottaz, R. Stalker, and I. Smith. Constraint solving and preference activation for interactive design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 12:13–27, 1998.
13. S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In AAAI 90, pages 25–32.
14. B. O'Sullivan. Constraint-Aided Conceptual Design. PhD thesis, Department of Computer Science, University College Cork, Ireland, July 1999. (Also Professional Engineering Publishing, 2001).
15. B. O'Sullivan. Interactive Constraint-Aided Conceptual Design. AIEDAM, 16(4):303–328, 2002.
16. G. Pahl and W. Beitz. Engineering Design: A systematic approach. Springer, 1995.
17. C. Petrie, H. Jeon, and M.R. Cutkosky. Combining constraint propagation and backtracking for distributed engineering. In Workshop on Non-Standard Constraint Processing, ECAI 96, pages 84–94.
18. S. Y. Reddy, K.W. Fertig, and D. E. Smith. Constraint management methodology for conceptual design tradeoff studies. In Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, August 1996. Irvine, California.
19. D. Sabin and R. Weigel. Product configuration frameworks – a survey. IEEE Intelligent Systems and their applications, 13(4):42–49, July–August 1998.
20. D. Serrano. Constraint Management in Conceptual Design. PhD thesis, MIT, 1987.
21. S. Shimizu and M. Numao. Constraint-based design for 3D shapes. Artificial Intelligence Journal, 91:51–69, 1997.

Incremental Induction of Classification Rules for Cultural Heritage Documents

Teresa M.A. Basile, Stefano Ferilli, Nicola Di Mauro, and Floriana Esposito

Dipartimento di Informatica - Università degli Studi di Bari
via E. Orabona, 4 – 70125 Bari - Italia
{basile, ferilli, nicodimauro, esposito}@di.uniba.it

Abstract. This work presents the application of a first-order logic incremental learning system, INTHELEX, to learn rules for the automatic identification of a wide range of significant document classes and their related components. Specifically, the material includes multi-format cultural heritage documents concerning European films from the 20's and 30's provided by the EU project COLLATE. Incrementality plays a key role when the set of documents is continuously augmented. To ensure that there is no performance loss with respect to classical one-step systems, a comparison with Progol was carried out. Experimental results prove that the proposed approach is a viable solution, for both its performance and its effectiveness in the document processing domain.

1 Introduction

Many important historic and cultural sources, which constitute a major part of our cultural heritage, are fragile and distributed in various archives, which still lack effective and efficient technological support for cooperative and collaborative knowledge working. The IST-1999-20882 project COLLATE (Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material) aims at developing a WWW-based *collaboratory* [6] for archives, researchers and end-users working with digitized historic/cultural material (URL: `http://www.collate.de`). The chosen sample domain includes a large corpus of multi-format documents concerning rare historic film censorship from the 20's and 30's provided by three major European film archives, specifically, Deutsches Film Institut (DIF), Film Archive Austria (FAA) and Národni Filmový Archiv (NFA). In-depth analysis and comparison of such documents can give evidence about different film versions and cuts, and allow to restore lost or damaged films, or to identify actors and film fragments of unknown origin. The COLLATE system aims at providing suitable task-based interfaces and knowledge management tools to support film experts' individual work and collaboration in analyzing, indexing, annotating and interlinking such documents. Continuously integrating valuable knowledge about the cultural, political and social contexts into its digital data and metadata repositories, it will provide improved content-based functionality to better retrieve and interpret such a material.

In this environment, the automatic induction of rules that are able to recognize the document classes and their significant components in order to provide them to the film

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 915-923, 2004.

experts would be very helpful. In particular, the complexity of the available documents layout structure suggests the use of symbolic (first-order logic) descriptions and techniques. Good results in this field would be a strong motivation to extend the application of the presented techniques to other kinds of documents of interest in the field of office automation. In this perspective, one possible application would be the classification of new incoming documents, e.g. invoices/letters/advertisements, and store them by type in a database. This led us to try applying the INTHELEX learning system to this domain.

The following section presents INTHELEX, an incremental learning system, along with its reasoning strategies; then, Section 3 reports the experimental results obtained on COLLATE documents and the comparison with the batch system Progol [10], that differently from INTHELEX performs learning in one step, i.e. it requires all the information needed for carrying out its task to be available when the learning process starts. Lastly, Section 4 draws some conclusions.

2 Incremental Learning with INTHELEX

Automatic revision of logic theories is a complex and computationally expensive task. Incremental learning is necessary when incomplete information is available at the time of initial theory generation, which is very frequent in real-world situations. Hence, the need for incremental models to complete and support the classical batch ones, that perform learning in one step and thus require the whole set of observations to be available since the beginning. Such a consideration, among others on the incremental learning systems available in the literature led to the design and implementation of INTHELEX[1] (INcremental THEory Learner from EXamples) [3], whose most characterizing features are in its incremental nature, in the reduced need of a deep background knowledge, in the exploitation of negative information and in the peculiar bias on the generalization model, which reduces the search space and does not limit the expressive power of the adopted representation language.

2.1 The Inductive Core

INTHELEX is a learning system for the induction of *hierarchical* logic theories from examples: it is *fully incremental* (in addition to the possibility of refining a previously generated version of the theory, learning can also start from an empty theory); it is based on the *Object Identity* assumption (terms, even variables, denoted by different names within a formula must refer to different objects) and learns theories expressed as sets of Datalog^{OI} clauses [11] from positive and negative examples; it can learn simultaneously *multiple concepts*, possibly related to each other (recursion is not allowed); it retains all the processed examples, so to guarantee validity of the learned theories on all of them; it is a *closed loop* learning system (i.e., a system in which feedback on performance is used to activate the theory revision phase [1]).

[1] It is currently available in binary format for i586 DOS-based platforms (`http://lacam.di.uniba.it:8000/systems/inthelex/`)

The learning cycle performed by INTHELEX can be described as follows. A set of examples of the concepts to be learned, possibly selected by an expert, is provided by the environment. This set can be subdivided into three subsets, namely training, tuning, and test examples, according to the way in which examples are exploited during the learning process. Specifically, training examples, previously classified by the expert, are stored in the base of processed examples, then they are exploited to obtain a theory that is able to explain them. Such an initial theory can also be provided by the expert, or even be empty. Subsequently, the validity of the theory against new available examples, also stored in the example base, is checked by taking the set of inductive hypotheses and a tuning/test example as input and producing a decision that is compared to the correct one. In the case of incorrectness on a tuning example, the cause of the wrong decision can be located and the proper kind of correction chosen, firing the theory revision process. Specifically, INTHELEX incorporates two inductive refinement operators to revise the theory, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative examples. In this way, tuning examples are exploited incrementally to modify incorrect hypotheses according to a data-driven strategy. Test examples are exploited just to check the predictive capabilities of the theory, intended as the behavior of the theory on new observations, without causing a refinement of the theory in the case of incorrectness.

Whenever a new example is taken into account, it is stored in the historical memory of all past examined examples and the current theory is checked against it. If it is positive and not covered, generalization must be performed. One of the clauses defining the concept the example refers to is chosen by the system for generalization. The set of generalizations of this clause and the example is computed, by taking into account a number of parameters that restrict the search space according to the degree of generalization to be obtained and the computational budget allowed. If one of such generalizations is consistent with all the past negative examples, then it replaces the chosen clause in the theory, or else a new clause is chosen to compute generalization. If no clause can be generalized in a consistent way, the system checks if the example itself, with the constants properly turned into variables, is consistent with the past negative examples. If so, such a clause is added to the theory, or else the example itself is added as an exception.

If the example is negative and covered, specialization is needed. Among the theory clauses occurring in the derivation of the example, INTHELEX tries to specialize one at the lowest possible level in the dependency graph by adding to it one (or more) positive literal(s), which characterize all the past positive examples and can discriminate them from the current negative one. Again, parameters that bound the search for the set of literals to be added are considered. In case of failure on all of the clauses in the derivation, the system tries to add the negation of a literal, that is able to discriminate the negative example from all the past positive ones, to the clause related to the concept the example is an instance of. If this fails too, the negative example is added to the theory as an exception. New incoming observations are always checked against the exceptions before applying the rules that define the concept they refer to.

2.2 Multistrategy Learning

Another peculiarity in INTHELEX is the integration of multistrategy operators that may help in the solution of the theory revision problem by pre-processing the incoming information, according to the theoretical framework for integrating different learning strategies known as Inferential Learning Theory [9]. Namely, deduction is exploited to fill observations with information that is not explicitly stated, but is implicit in their description, and hence refers to the possibility of better representing the examples and, consequently, the inferred theories. Conversely, abduction aims at completing possibly partial information in the examples (adding more details), whereas abstraction removes superfluous details from the description of both the examples and the theory. Thus, even if with opposite perspectives, both aim at reducing the computational effort required to learn a correct theory with respect to the incoming examples.

INTHELEX requires the observations to be expressed only in terms of the set of predicates that make up the description language for the given learning problem. To ensure uniformity of the example descriptions, such predicates have no definition. Nevertheless, since the system is able to handle a hierarchy of concepts, combinations of these predicates might identify higher level concepts that is worth adding to the descriptions in order to raise their semantic level. For this reason, INTHELEX implements a saturation operator that exploits deduction to recognize such concepts and explicitly add them to the examples description. The system can be provided with a Background Knowledge, supposed to be correct and hence not modifiable, containing (complete or partial) definitions in the same format as the theory rules. This way, any time a new example is considered, a preliminary saturation phase can be performed, that adds the higher level concepts whose presence can be deduced from such rules by subsumption and/or resolution. In particular, the generalization model of implication under Object Identity is exploited [5]. Differently from abstraction (see next), all the specific information used by saturation is left in the example description. Hence, it is preserved in the learning process until other evidence reveals it is not significant for the concept definition, which is a more cautious behaviour.

Abduction was defined by Peirce as hypothesizing some facts that, together with a given theory, could explain a given observation. According to the framework proposed in [7], an *abductive logic theory* is made up by a normal logic program [8], a set of *abducibles* and a set of *integrity constraints*. Abducibles are the predicates about which assumptions (*abductions*) can be made: They carry all the incompleteness of the domain (if it were possible to complete these predicates then the theory would be correctly described). Integrity constraints (each corresponding to a combination of literals that is not allowed to occur) provide indirect information about them. The proof procedure implemented in INTHELEX corresponds, intuitively, to the standard Logic Programming derivation suitably extended in order to consider abducibles.

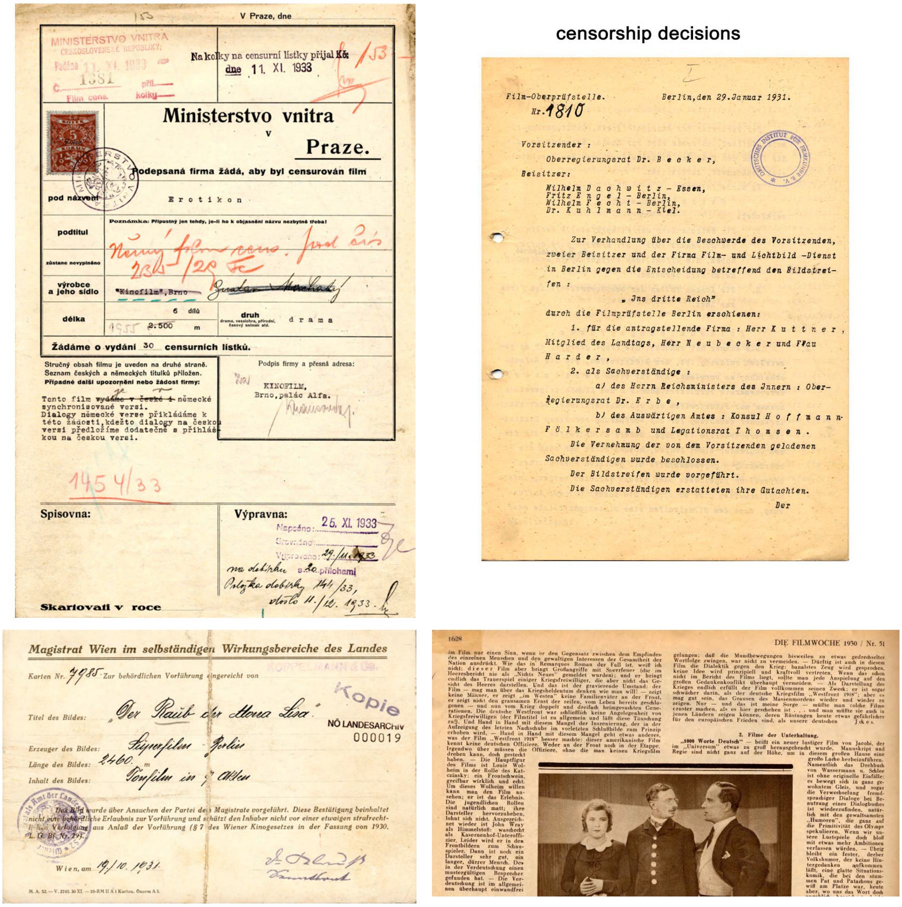
Abstraction is a pervasive activity in human perception and reasoning. When we are interested in the role it plays in Machine Learning, inductive inference must be taken into account as well. The exploitation of abstraction concerns the shift from the language in which the theory is described to a higher level one. According to the framework proposed in [13], concept representation deals with entities belonging to three different levels. Concrete objects reside in the *world*, but any observer's access to it is mediated by his *perception* of it. To be available over time, these stimuli must be memorized in an organized *structure*, i.e. an *extensional* representation of the perceived world. Finally, to reason about the perceived world and communicate with other agents, a *language* is needed, that describes it *intensionally*. Abstraction takes place at the world-perception level by means of a set of operators, and then propagates to higher levels, where it is possible to identify operators corresponding to the previous ones. An abstraction theory expresses such operators, that allow the system to replace a number of components by a compound object, to decrease the granularity of a set of values, to ignore whole objects or just part of their features, and to neglect the number of occurrences of some kind of object. In INTHELEX the abstraction theory must be given, and the system automatically applies it to the learning problem at hand before processing the examples.

3 Experimental Results on the COLLATE Dataset

Supported by previous successful application to the paper document processing domain [12], the symbolic learning system INTHELEX has been applied to learn rules for the automatic identification of a wide range of significant COLLATE classes and their related components, to be used for indexing/retrieval purposes and to be submitted to the users for annotation. A challenge comes from the low layout quality and standard of the material, which causes a considerable amount of noise in its description (see Figure 1). The layout quality is often affected by manual annotations, stamps that overlap to sensible components, ink specks, etc. As to the layout standard, many documents are typewritten sheets, that consist of all equally spaced lines in Gothic type. Such a situation requires the system to be flexible in the absence of particular layout components due to the typist's style, and to be able to ignore layout details that are meaningless or superfluous to the identification of the interesting ones.

The dataset consisted of 29 documents for the class `faa_registration_card` (a certification that the film has been approved for exhibition in the present version by the censoring authority), 36 ones for the class `dif_censorship_decision` (decision whether a film could or could not, and in which version, be distributed and shown throughout a Country), 24 ones for the class `nfa_cen_dec_model_a` and 13 for the class `nfa_cen_dec_model_b` (both these classes represent a list of intertitles needed to check whether a film shown in the cinema was the same as the one examined by the censorship office). Other 17 reject documents were obtained from newspaper articles. Note that the symbolic method adopted allows the trainer to specifically select prototypical examples to be included in the learning set. This explains why theories with good predictiveness can be obtained even from fewer observations.

The first-order descriptions of such documents, needed to run the learning system, were automatically generated by the system WISDOM++ [4]. Starting from scanned images, such a system is able to identify the layout blocks that make up a paper document along with their type and relative position.


censorship decisions

153

V Praze, dne

MINISTERSTVO VNITRA
ČESKOSLOVENSKÉ REPUBLIKY:
Podáno 11. XI. 1933
č. 1381 příl. — kolky —
Film cens.

Na kolky na censurní lístky přijal Kč: 153
dne 11. XI. 1933

Ministerstvo vnitra v Praze.

Podepsaná firma žádá, aby byl censurován film

pod názvem: Erotikon

podtitul — Poznámka: Přípustný jen tehdy, je-li ho k objasnění názvu nezbytně třeba!

zůstane nevyplněno — Němý film cens. 236/29

výrobce a jeho sídlo: "Kinofilm", Brno ~~Gustav Machatý~~

délka: 6 dílů, ~~1955~~ 2.500 m — druh (drama, veselohra, přírodní, časový snímek atd.): drama

Žádáme o vydání 30 censurních lístků.

Stručný obsah filmu je uveden na druhé straně.
Seznam českých a německých titulků přiložen.
Případné další upozornění nebo žádost firmy:

Tento film ~~vydáme v české i~~ je v německé synchronisované versi.
Dialogy německé verse přikládáme k této žádosti, kdežto dialogy na českou versi předložíme dodatečně s přihláškou na českou versi.

Podpis firmy a přesná adresa:
KINOFILM,
Brno, palác Alfa.

1454/33

Spisovna:

Výpravna:
Napsáno: 25. XI. 1933
Srovnáno:
Vypraveno: 29/11. 1933
na dobírku s 20 přílohami
Položka dobírky 144/33,
došlo 11./12. 1933.

Skartovati v roce

I

Film-Oberprüfstelle. Berlin, den 29. Januar 1931.
Nr. 1810

DEUTSCHES INSTITUT FÜR FILMKUNDE E.V.

Vorsitzender:
Oberregierungsrat Dr. Becker,
Beisitzer:
Wilhelm Dachwitz - Essen,
Fritz Engel - Berlin,
Wilhelm Fecht - Berlin,
Dr. Kuhlmann - Kiel.

Zur Verhandlung über die Beschwerde des Vorsitzenden, zweier Beisitzer und der Firma Film- und Lichtbild -Dienst in Berlin gegen die Entscheidung betreffend den Bildstreifen:

„Ins dritte Reich"

durch die Filmprüfstelle Berlin erschienen:

1. für die antragstellende Firma: Herr Kuttner, Mitglied des Landtags, Herr Neubecker und Frau Harder,
2. als Sachverständige:
a) des Herrn Reichsministers des Innern: Oberregierungsrat Dr. Erbe,
b) des Auswärtigen Amtes: Konsul Hoffmann-Fölkersamb und Legationsrat Thomsen.

Die Vernehmung der von dem Vorsitzenden geladenen Sachverständigen wurde beschlossen.

Der Bildstreifen wurde vorgeführt.

Die Sachverständigen erstatteten ihre Gutachten.

Der

Magistrat Wien im selbständigen Wirkungsbereiche des Landes

Karten Nr. 7985 Zur behördlichen Vorführung eingereicht von KOPPELMANN & Co.

Kopie

NÖ LANDESARCHIV
000019

Titel des Bildes: „Der Raub der Mona Lisa"
Erzeuger des Bildes: Superfilm - Berlin
Länge des Bildes: 2460 m
Inhalt des Bildes: Tonfilm in 9 Akten

Das Bild wurde über Ansuchen der Partei dem Magistrate vorgeführt. Diese Bestätigung beinhaltet nicht eine behördliche Erlaubnis zur Vorführung und schützt den Inhaber nicht vor einer etwaigen strafrechtlichen Verfolgung aus Anlaß der Vorführung (§ 7 des Wiener Kinogesetzes in der Fassung von 1930, L. G. Bl. Nr. 29).

Magistrats Amt der Landeshauptstadt Wien

Wien, am 19/10. 1931.

M. A. 52. — V. 2745. 30 XI. — 10-RM II A 1 Karton. Önorm A 5.

1628 DIE FILMWOCHE 1930 / Nr. 51

im Film nur einen Sinn, wenn er den Gegensatz zwischen dem Empfinden des einzelnen Menschen und den gewaltigen Interessen der Gesamtheit der Nation ausdrückt. Wie das in Remarques Roman der Fall ist, weiß ich nicht: dieser Film aber bringt Großangriffe mit Sperrfeuer (die im Heeresbericht nie als „Nichts Neues" gemeldet wurden); und er bringt endlich das Trauerspiel einiger Kriegsfreiwilliger, die aber nicht das Gesicht des Heeres darstellen. Und das ist der gravierende Umstand: der Film — mag man über das Kriegsheldentum denken wie man will! — zeigt keine Männer, er zeigt „im Westen" keine Familienväter an der Front, er zeigt nicht den grausamen Ernst der reifen, vom Leben bereits geschlagenen — und nun vom Krieg doppelt und dreifach heimgesuchten Generationen. Die deutsche Westfront war schließlich keine Angelegenheit von Kriegsfreiwilligen (der Filmtitel ist zu allgemein und läßt diese Täuschung zu!). Und Hand in Hand mit diesem Mangel der Inszenierung, der in der Aufzeigung des letzten Nachschubs im vorletzten Schlußbilde zum Prinzip erhoben wird, — Hand in Hand mit diesem Mangel geht etwas anderes, was der Film „Westfront 1918" besser machte: dieser amerikanische Film kennt keine deutschen Offiziere. Weder an der Front noch in der Etappe. Irgendwo über müssen die Offiziere, ohne die man keinen Kriegsfilm drehen kann, doch gesteckt haben. — Die Hauptfigur des Films ist Louis Wolheim in der Rolle des Katczinsky; ein Frontschwein, greifbar wirklich und echt. Um dieses Wolheim willen kann man den Film ansehen; er ist das Erlebnis. Die jugendlichen Rollen sind natürlich matt; ihre Darsteller hervorzuheben, lohnt sich nicht. Ausgezeichnet wieder ist John Wray als Himmelstoß: waschecht als Kasernenhof-Unteroffizier. Leider wird er in den Frontbildern zum Schauspieler. Dann ist noch ein Darsteller sehr gut, ein langer, dürrer Mensch. Der in der Verdeutschung einen mustergültigen Besprecher gefunden hat. — Die Verdeutschung ist im allgemeinen überhaupt einwandfrei gelungen: daß die Mundbewegungen bisweilen zu etwas gedrechselter Wortfolge zwingen, war nicht zu vermeiden. — Dürftig ist auch in diesem Film die Dialektik gegen den Krieg: banalstes Zeug wird gesprochen, keine Idee wird prinzipiell und kraftvoll vertreten. Wenn das schon nicht im Bericht des Films liegt, sollte man jede Anspielung auf den großen Gedankenkonflikt überhaupt vermeiden. — Als Darstellung des Krieges endlich erfüllt der Film vollkommen seinen Zweck; er ist sogar schwächer darin, als der deutsche Kriegsfilm „Westfront 1918"; aber es mag gut sein, das Grausen des Massenmordens wieder und wieder zu zeigen. Nur — und das ist meine Sorge — müßte man solche Filme ernster machen, als es hier geschehen ist . . . und man müßte sie auch in jenen Ländern zeigen können, deren Rüstungen heute etwas gefährlicher für den europäischen Frieden sind, als unsere deutschen. Jckes.

2. Filme der Unterhaltung.

„1000 Worte Deutsch" — heißt ein neuer lustiger Film von Jacobi, der im „Universum" etwas zu groß herausgebracht wurde. Manuskript und Regie sind nicht ganz auf der Höhe, um in diesem großen Hause eine große Lache herbeizuführen. Namentlich das Drehbuch von Wassermann u. Schlee ist ohne originelle Einfälle; es bewegt sich in ganz gewohntem Gleis, und sogar die Verwechselung fremdsprachiger Dialoge bei Benutzung eines Dialogbuches ist wiederzufinden, natürlich mit den gewaltsamsten „Humoren", die ganz auf die Primitivität des Olymps spekulieren. Wenn wir unsere Lustspiele doch bloß mit etwas mehr Ambitionen verfassen würden. — Übrig bleibt ein fester, derber Volkshumor, der keine Hintergedanken aufkommen läßt, eine glatte Situationskomik, die bei den stummen Pat und Patachons gewiß am Platze war, heute aber, wo uns das Wort doch

Fig. 1. Sample COLLATE documents

Each document was then described in terms of its composing layout blocks, along with their size (height and width), position (horizontal and vertical), type (text, line, picture and mixed) and relative position (horizontal/vertical alignment, adjacency).
The description length of the documents ranges between 40 and 379 literals (144 on average) for class `faa_registration_card`, between 54 and 263 (215 on average) for class `dif_censorship_decision`; between 105 and 585 (269 on average) for class `nfa_cen_dec_model_a` and between 191 and 384 (260 on average) for class `nfa_cen_dec_model_b`.

Each document was considered as a positive example for the class it belongs, and as a negative example for the other classes to be learned; reject documents were considered as negative examples for all classes. Definitions for each class were learned, starting from the empty theory and with all the negative examples at the beginning (in order to simulate a batch approach), and their predictive accuracy was tested according to a 10–fold cross validation methodology, ensuring that each fold contained the same proportion of positive and negative examples.

As regards the rules learned by INTHELEX, Figure 2 shows a definition for the classification of documents belonging to dif_censorship_decision class. It is interesting to note that it is straightforwardly understandable by humans. Specifically, the English translation of the concept expressed by this rule is "*a document belongs to this class if it has long length and short width, it contains three components in the upper-left part, all of type text and having very short height, two of which are medium large and one of these two is on top of the third* ". This feature was greatly appreciated by experts in charge of working with the processed documents, and was one of the goals we aimed to.

```
class_dif_cen_decision(A) :-
   image_lenght_long(A),image_width_short(A),
   part_of(A,B),
   width_medium_large(B),height_very_very_small(B),
   type_of_text(B),pos_left(B),pos_upper(B),
   part_of(A,C),
   height_very_very_small(C),type_of_text(C),
   pos_left(C),pos_upper(C),
   on_top(C,D),
   width_medium_large(D),height_very_very_small(D),
   type_of_text(D),pos_left(D),pos_upper(D).
```

Fig. 2. Examples of Learned Definitions

Two remarks are worth for this class: first, the features in the above description are common to all the learned definitions in the 10 folds, which explains why the performance of the system on this class of documents is the best of all; second, starting with descriptions whose average length was 215, the average number of literals in the learned rules is just 22.

Experiments were run not only with INTHELEX, but also with the state-of-the-art system Progol. The aim was checking that there is no loss in performance using the incremental technique instead of the batch one. This would allow to safely exploit the incremental approach in domains characterized by a continuous flow of new documents. Table 1 reports the statistics regarding the performance of the two exploited approaches, averaged on the 10 folds, of the classification process in this environment as regards number of clauses that define the concept Clauses, Accuracy on the test set (expressed in percentage) and Runtime (in seconds).

The difference in computational time between the two systems is noteworthy, confirming that the incremental approach should be more efficient than the batch one (since it has to just revise theories stepwise, instead of learning them from scratch). On the contrary, predictive accuracy seems very similar, which suggested to perform a statistical test for assessing its significance.

Thus, to better assess the goodness of INTHELEX, the performance of the system on these datasets in the above 10-fold cross validation was compared, according to a paired *t*-test [2], to that obtained by the Progol batch system. The aim was to evaluate the difference in effectiveness of the rules induced by the two systems according to the predictive accuracy metric. Table 2 reports the results of such a comparison, requiring a significance level of α = 0.995. In hypothesis testing, the significance level of a test is the probability of incorrectly rejecting the null hypothesis. In our experiment the null hypothesis is: "*the two systems are equally performing*". Here, we want some guarantee that the two systems are comparable, in order to apply the incremental one in real word domains, such as office automation. In other words, we want be sure with a high probability that there are no differences among the two systems. Thus, we have chosen an high significance, that assures the systems equality with a very small margin to make a mistake on evaluating their performance. The test revealed no statistically significant differences in predictive accuracy among the systems in the classification task in the cultural heritage material environment.

Table 1. Statistics for Document Classification

	Clauses		Runtime (sec.)		Accuracy (%)	
	Progol	INTHELEX	Progol	INTHELEX	Progol	INTHELEX
DIF	1.0	1.0	687.70	17.13	99.17	99.17
FAA	3.6	3.5	3191.98	334.05	95.83	94.17
NFA_A	4.6	2.8	1558.90	87.71	95.73	93.92
NFA_B	1.0	1.7	359.67	92.05	97.63	97.56

Table 2. INTHELEX – Progol Comparison

	Accuracy (%)		
	Progol	INTHELEX	t-value
DIF	99.17	99.17	0
FAA	95.83	94.17	0.80
Mod_A	95.73	93.92	0.57
Mod_B	97.63	97.56	0.03

The experimental results show that INTHELEX is able to learn theories with performance comparable to the batch systems ones. This seems interesting, since the incremental setting implies having, at any moment, only a limited vision of the domain. Conversely, batch systems such as Progol can consider the whole knowledge available since the beginning of the learning process, so having a general vision of the knowledge available. Thus, such a feature should in principle result in a better predictive accuracy of theories learned by the latter with respect to those provided by the former.

4 Conclusion and Future Work

This paper presented experimental results proving the benefits that the addition of the incremental learning system INTHELEX in the architecture of the EU project COLLATE can bring, in order to learn rules for automatic classification and interpretation of cultural heritage documents. The domain is particularly challenging because of the low layout quality and standard of the material, which can represent a good testbed in the perspective of applying the same techniques to the field of office automation, where the continuous flow of new documents makes incrementality a necessary feature. INTHELEX works on symbolic (first-order logic) representations, that proved very powerful in describing a complex environment such as the COLLATE one. This was confirmed by a comparison with the batch learning system Progol on the predictive accuracy metric, revealing that there are generally no statistically significant differences among the two systems.

The presented experiments were carried out by exploiting only pure induction. Nevertheless, the multistrategy features provided by INTHELEX could probably further improve the performance. Thus, future work will concern performing new experiments aimed at comparing multistrategy learning with respect to the baseline results presented in this paper.

Acknowledgement. This work was partially funded by the EU project IST-1999-20882 COLLATE "Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material".

References

1. J. M. Becker. Inductive learning of decision rules with exceptions: Methodology and experimentation. B.s. diss., Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1985.
2. Thomas G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895--1923, 1998.
3. F. Esposito, G. Semeraro, N. Fanizzi and S. Ferilli. Multistrategy Theory Revision: Induction and abduction in INTHELEX. *Machine Learning Journal*, 38(1/2):133--156, 2000.
4. F. Esposito, D. Malerba and F.A. Lisi. Machine learning for intelligent processing of printed documents. *Journal of Intelligent Information Systems*, 14(2/3):175--198, 2000.
5. F. Esposito, N. Fanizzi, S. Ferilli and G. Semeraro. Refining logic theories under oi-implication. In Z. W. Ras and S. Ohsuga, editors, *Foundations of Intelligent Systems*, No. 1932 in Lecture Notes in Artificial Intelligence, pages 109--118. Springer-Verlag, 2000.
6. R.T. Kouzes, J.D. Myers, and W.A. Wulf. Collaboratories: Doing science on the internet. *IEEE Computer*, 29(8), 1996.
7. E. Lamma, P. Mello, F. Riguzzi, F. Esposito, S. Ferilli and G. Semeraro. Cooperation of abduction and induction in logic programming. In A. C. Kakas and P. Flach, editors, *Abductive and Inductive Reasoning: Essays on their Relation and Integration*. Kluwer, 2000.
8. J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, second edition, 1987.
9. R.S. Michalski. Inferential theory of learning. developing foundations for multistrategy learning. In R.S. Michalski and G. Tecuci, editors, *Machine Learning. A Multistrategy Approach*, volume IV, pages 3--61. Morgan Kaufmann, San Mateo, CA, 1994.
10. S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245--286, 1995.
11. G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi and S. Ferilli. A logic framework for the incremental inductive synthesis of Datalog theories. In N.E. Fuchs, editor, *Proceedings of 7th International Workshop on Logic Program Synthesis and Transformation* LOPSTR97, volume 1463 of *LNCS*, pages 300--321. Springer, 1998.
12. 12.G. Semeraro, N. Fanizzi, S. Ferilli and F. Esposito. Document classification and interpretation through the inference of logic-based models. In P. Constantopoulos and I.T. Sølvberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in Lecture Notes in Computer Science, pages 59--70. Springer-Verlag, 2001.
13. J. D. Zucker. Semantic abstraction for concept representation and learning. In R. S. Michalski and L. Saitta, editors, *Proceedings of the 4th International Workshop on Multistrategy Learning*, Desenzano del Garda, Italy, 1998.

Applying Multi-class SVMs into Scene Image Classification

Jianfeng Ren, Yuntao Shen, Songhui Ma, and Lei Guo

Department of Automatic Control, North Western Polytechnic University
Xi' an, china, 710072
rjjff@163.com lguo@nwpu.edu.cn

Abstract. Grouping images into semantically meaningful categories using the low-level visual features is a challenging and important problem in content-based image retrieval and other applications. In this paper, we show a specific high-level classification problem (scene images classification) using the low level features such as representative colors and Gabor textures. Based on the low level features, we introduce the multi-class SVMs to merge these features with the final goal to classify the different scene images. Experimental results show our method is promising.

1 Introduction

Image classification commonly means that grouping images into semantically meaningful categories based on the available training data. However, although some work [1,2,3] has been done in this field, little results have been obtained to satisfy user's expectations. Image classification is still a challenging and important problem in compute vision recently.

In the feature extraction, many methods have been developed such as the histogram [4] and texture [5]. The disadvantage of [4] is that the dimensionality of histogram is too high. So in this paper, we provide a new method to compute its histogram called dominant color for representing the images. In [5], the input of classifier is the raw pixels of the segmented images rather than the extracted features. Therefore, this method has the obvious drawback that is computationally expensive. So the texture features are provided in this paper by using Gabor-filters which can capture the texture's edges and directions very effectively.

From Bayes classifier to neural networks, there are many possible choices for an appropriate classifier. Among these, support vector machines (SVMs) would appear to be a good candidate because of their ability to generalize in high-dimensional spaces without the need to add a prior knowledge. The appeal of SVMs is based on their strong connection to the underlying statistical learning theory. That is, an SVM is an appropriate implementation of the structural risk minimization method [6]. For several pattern classification applications [7,8], SVMs have been shown to provide better generalization performance than traditional techniques such as neural networks [9].

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 924-934, 2004.

For the convenience, the object of our research has been narrowed into scene images from the benchmark data which are available at http://www.project.-minerva.ex.ac.uk. Therefore, the aim of this paper is to illustrate the potential of SVMs in scene classification. According to human observation, the objects of the images from the benchmark data may be brick, grass, leave, pebble, sky, road and tree. In order to classify the above objects efficiently, we solve the problem by using multi-class SVMs instead of binary SVMs classifier. Firstly the images are segmented into five fixed regions, which are shown in figure 1, just because a whole image can consist of two or more objects. If the segmented images still consist of more than one object, we should discard them from our experimental database. Then the dominant color histogram and Gabor-based textures are extracted based on the segmented images. Finally the features are applied in the multi-class SVMs with the goal to classify the images.

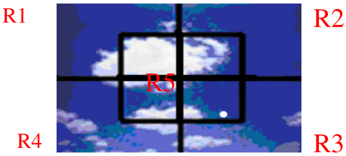


Fig. 1. Image decomposition

This paper is organized as the follows: in section 2,we will briefly introduce the previous work on image classification. In section 3, overview of SVMs and multi-class SVMs will be provided. We will describe the feature extraction such as dominant color histogram and Gabor-based on textures in section 4. In section 5, Experimental results will be provided and discussed. Finally, conclusions will be given in section 6.

2 Related Work

Several attempts at classifying images using low-level features and different approaches of classifiers have been made.

In [10], Martin Szummer showed how high-level scene properties can be inferred from classification of low-level image features, specially for the indoor-outdoor classification. They studied three different low-level features: (1) histograms in the Ohta color space (2) multiresolution, simultaneous autoregressive model parameters (3) coefficients of a shift-invariant DCT. And they apply three different classifiers and make comparisons based on the above features. Finally they conclude that relatively simple classifiers (k-nearest neighbors) performed better than the more sophisticated neural networks and mixture of expert classifiers.

In [11], Aditya Vailaya studied the classification of city images and landscapes which can be solved from simple low-level features such as color histogram, color coherence vector, DCT coefficient, edge direction histogram and edge direction. A weighted *k-NN* classifier is used for the classification which results in an accuracy of

93.9% when evaluated on an image database 2716 images using the leave-one-out method.

However, research work [10,11] mainly focuses on two-class classification problem. In fact, the real world images can be classified more than two classes. Accordingly, Markos Markou and Sameer Singh [12] began to conduct research on multi-class image classification based on neural network. In this paper, we conduct research on the same image database from [12] based on SVMs instead of neural networks. Experimental results show that our provided method outperforms the method provided in [12].

3 Support Vector Machines

3.1 Optimal Separating Hyperplanes

We give in this section a brief introduction to SVMs. Let $(x_i, y_i)_{1 \le i \le N}$ be a set of training samples, each sample $x_i \in R^d$, d being the dimension of the input space, belongs to a class labeled by $y_i \in \{+1, -1\}$. The aim is to define a hyperplane which divides the set of samples such that all the points with the same label are the same side of the hyperplane.

$$y_i(wx_i + b) \succ 0 \qquad \mathrm{i} = 1, \ldots, \mathrm{N} \tag{1}$$

This amounts to finding w and b so that if there exists a hyperplane satisfying (1), the set is said to be linearly separable. In this case, it is always possible to rescale w and b so that $\min\limits_{1 \le i \le N} y_i(wx_i + b) \ge 1 \quad \mathrm{i} = 1, \cdots, \mathrm{N}$ i.e., so that the distance between the closest point to the hyperplane is $1/\|w\|$. Then, (1) becomes

$$y_i(wx_i + b) \ge 1 \qquad \mathrm{i} = 1, \ldots, \mathrm{N} \tag{2}$$

Among the separating hyperplanes, the one for which the distance to the closet point is maximal is called optimal separating hyperplane *(OSH).* Since the distance to the closest point is $1/\|w\|$, finding the OSH amounts to minimizing $\|w\|^2$ under constraints (2).

The quantity $2/\|w\|$ is called the margin, and thus the OSH is separating hyperplane which maximizes the margin. The margin can be seen as a measure of the generalization ability: the larger margin, the better generalization is expected [6].

Since $\|w\|^2$ is convex, minimizing it under linear constraints (2) can be achieved with Lagrange multipliers. If we denote by $\alpha = (\alpha_1, \cdots, \alpha_N)$ the N nonnegative Lagrange multipliers associated with constraints (2). Our optimization problem amounts to maximizing where $\alpha_i \ge 0$ and under constraints $\sum_{i=1}^{N} y_i \alpha_i = 0$.

$$w_0 = \sum_{i=1}^{N} \alpha_i^0 y_i x_i \tag{3}$$

This can be achieved by use of standard quadratic programming method. However, for large learning tasks with many training samples, solving the quadratic optimization programs with bound constraints and one linearly equality constraints quickly become intractable in the memory and time requirements. Accordingly, a new fast algorithm for SVM called sequential minimal optimization [13] was developed in order to make SVM more practical. In this paper, we also adopted this algorithm into our research work.

Once the vector $\alpha^0 = (\alpha_1^0, \cdots, \alpha_N^0)$ solution of the maximization problem (3) has been found, the OSH (w_0, b_0) has the following expansion:

$$w_0 = \sum_{i=1}^{N} \alpha_i^0 y_i x_i \tag{4}$$

The support vectors are the points for which $\alpha_i^0 > 0$ satisfy (2) with equality.

Considering the expansion (4) of w_0, the hyperplane decision function can be thus written as

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i^0 y_i x_i \bullet x + b_0 \right) \tag{5}$$

3.2 Linearly Non-separable Case

When the data is not linearly separable, we introduce slack variables $(\xi_1, \cdots, \xi_N)$ with $\xi_i \geq 0$ to allow the possibility of examples that violate (2)

$$y_i(w \bullet x_i + b) \geq 1 - \xi_i \qquad \mathrm{i} = 1, \cdots, \mathrm{N} \tag{6}$$

The purpose of the variables ξ_i is to allow misclassified points, which have their corresponding $\xi_i > 1$. Therefore $\sum \xi_i$ is an upper bound on the number of training errors. The generalized OSH is then regarded as the solution of the following problem of minimizing subject to constraints (6) and $\xi_i \geq 0$.

$$\frac{1}{2} w \bullet w + C \sum_{i=1}^{N} \xi_i \tag{7}$$

The firs term is minimized to control the learning capacity as in the separable case; the purpose of the second term is to control the number of misclassified points. The parameter is chosen by the user, a larger C corresponding to assigning a higher penalty to errors.

3.3 Nonlinear Support Vector Machines

However, since it is unlikely that a general pattern recognition problem can actually be solved by linear classifier (separating hyperplane), the OSH needs to be augmented in order to allow for nonlinear decision surfaces. The basic idea is to map the data into a high dimension feature space through some nonlinear mapping $\Phi : R^d \to F$ and perform the above algorithm in F. Since the solution has the form, it is nonlinear in the original input variables.

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{N} y_i \alpha_i \phi(x_i) \bullet \phi(x) + b) \tag{8}$$

According to the Cover's theorem [14] states that such a multidimensional space can be transformed into a new feature space F where the patterns are linearly separable with a high probability, provided two conditions are satisfied: first the transformation is nonlinear and second, the dimensionality of the feature space is high enough. Accordingly, F usually needs to have a very high dimensionality in order to be linearly separable. And the solution is to replace all the occurrence of an inner product resulting from two mappings with the kernel function K defined as:

$$K(x) = \phi(x) \bullet \phi(y) \tag{9}$$

Conversely, given a symmetric positive kernel $K(x, y)$, Mercer's theorem [15] implies the existence of a mapping ϕ such that $K(x) = \phi(x) \bullet \phi(y)$. Then, without considering the mapping ϕ explicitly, a nonlinear SVM can be constructed by selecting the proper kernel. Table 1 summarizes the kernel functions for three common types of SVMs: Polynomial learning machines, radial basis function networks and two-layer perceptrons.

Table 1. Possible kernel functions and types of classifiers

Inner product kernel	Type of classifier
Polynomial kernel: $K(x, y) = (xy + 1)^d$	Polynomial learning machine
Gaussian kernel: $K(x, y) = \exp(-\frac{1}{2\sigma^2} \|x - y\|^2)$	Radial-basis function network
Tangent hyperbolic kernel: $K(x, y) = \tanh(xy - \Theta)$	Two-layer perceptron

3.4 Multi-class Support Vector Machines

SVMs are designed for binary classification. Various approaches have been developed in order to extend the standard SVMs to multi-class cases. The following strategies can be applied to build K-class classifiers utilizing binary SVM classifiers:

1. K one-against-rest classifiers
   The *ith* SVM will be trained with all of the examples in the *i*th class with positive labels, and all other examples with negative labels. We refer to SVMs trained in this way as l-v-r SVMs. The final output of the K l-v-r SVMs is the class that corresponds to the SVM with highest output value. This method suffers several drawbacks:(1)We have to relabel all the samples. (2) There is no bound on the generalization error for the 1-v-r SVM. (3) The training time of the standard method scales linearly with K
2. $K(K-1)/2$ one-against-one classifier
   For each possible pair of classes a binary classifier is calculated. Each classifier is trained on a subset of the training set containing only training examples of two involved classes. All classifiers are combined through a majority voting scheme to estimate the final classification. Here the class with maximal number of votes among all classifiers is the estimation.

In our paper, we only consider the latter case. Based on the one-against-one classifier, different schemes are developed for generating the multi-class classifier.

- Friedman [16]suggested a Max win algorithm: each 1-v-l classifier casts one vote for its preferred class, and the final result is the class with the most votes. Friedman shows circumstances in which this algorithm is Bayes optimal. Unfortunately this method does not have bounds on the generalization error.
- Pairwise coupling [17] trains $K(K-1)/2$ binary classifier, each of which can produce a pairwise probability. Pairwise coupling couples these pairwise probabilities into a common set of posterior probabilities. However, such method has also some drawbacks. When a sample $\bar{x}$ is classified by $K(K-1)/2$ classifiers, and at the same time, $\bar{x}$ doesn't belong to both of the two involved classes of this classifier, the probabilistic measures of $\bar{x}$ to the two classes are meaningless and maybe damage the coupling output of pariwise coupling.
- John C.Platt described a large margin DAGs for multi-class classification[18], in which he made use of the Decision Directed Acyclic Graph a learning architecture to combine many two-classifiers. It is called the Directed Acyclic Graph Support Vector Machines. The two-classifier SVM directed acyclic graph has $K(K-1)/2$ internal nodes and K leaves. Each node is a two-class SVM classifier of *i*th and *j*th classes. Given a test example x starting from the root node, the decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore we go through the path before each a leaf node that indicates the predicated class. The simple description of the algorithm is shown in figure 2.

4 Feature Extraction

In [4,5], only one low-level feature was adopted for representing the images, however, one low-level feature can not fully represent the image and will generate high error rate during the image classification. In section 5, some experimental results combined features will outperform better than single feature.

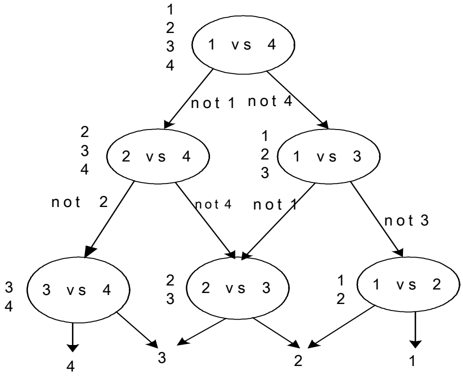


Fig. 2. The decision DAG for finding the best class out of four classes

4.1 Dominant Color Histogram

According to human vision perception, a small number of image colors are sufficient to express the color information. Therefore, in our method, we regard eight colors as the dominant colors. Eight colors are illustrated as following: $\{red, green, b\text{lue,yellow,magenta,cyan,black and white}\}$, which can be respectively mapped into corresponding points in the RGB space:

$$\{\{255,0,0\},\{0,255,0\},\{0,0,255\},\{255,255,0\},\{255,0,255\},\{0,255,255\},$$
$$(0,0,0\},\{255,255,255\}\}$$

In order to compute dominant color histogram, the colors in the image should be clustered based on the nearest neighbor algorithm.

4.2 Gabor-Based Features

Among the texture features, features computed from Gabor filtered images appear quite promising. Using the gabor filters designed in [18], an image is decomposed at K orientations and S scales. For a given image $I(x, y)$, its decomposed image at scale $m(m = 1, \cdots, S)$ and direction $n(n = 1, \cdots, K)$ is defined to be

$$W_{mn}(x, y) = \int I(x, y) g^*_{mn}(x - x_1, y - y_1) dx_1 dy_1 \qquad (10)$$

Where $W_{mn}(x, y)$ is the filtered image at a specific scale m and direction n. In this paper, we set S=4 and K=6. For each Gabor filter, we will compute two moments: the mean μ_{ij} and the standard deviation σ_{ij}, therefore, we will obtain the following vector $[\mu_{11}, \sigma_{11}, \cdots, \mu_{24}, \sigma_{24}]$ for representing the image

Based on the Gabor feature, SVM are adopted for classification. In fact, SVM may be as the same as the multichannel filtering. The operation performed by a kernel in an SVM is essentially the same of a channel in a multi-channel filtering method: the kernel computes the inner product $K(x, y = \phi(x)\phi(y)$ between the input x and the SVs y . Accordingly, the SVs play the role of a filter bank. As such, it should noted that an SVM can emphasize the correct separation of training data through the selection of filters (SVs), while, in a classical multi-channel filtering approach, the filter selection criteria are generally not directed to classification performance. In this paper, we first adopted the multi-channel filtering for feature extraction and then SVMs for classification.

5 Experimental Results

5.1 Experimental Setup

The images are from the benchmark data available at http://www.project.-minerva.ex.ac.uk. There are about 448 color images. And images of different natural scenes contain some objects such as grass, brick, pebbles, leaves, sky, road and trees. All images are stored in bitmap format with 24-bit color depth and a resolution of 720*512 pixel. For our research purpose, we convert all images into a resolution of 256*256 pixel and segmented images into five fixed regions according to figure 1. Therefore, we obtained about 2240 subimages with a resolution of 128*128.

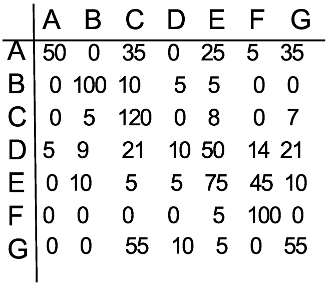
Then we asked for two students to classify the subimages into one of seven objects. If one image consists of more than one object, they were asked to discard it from the experimental database. Then we will obtain the following classification results: bricks (313), grass (254), leaves (297), pebbles (278), road (278), sky (234) and trees (251) and discard the remaining images from our experimental database. We select the about the bricks (150), grass (120), leaves (140), pebbles (130), roads (150), sky (105) and trees (125) as our training samples, and randomly select some images from the remaining images as the testing samples.

5.2 Experimental Results

In our research, we develop several experiments to study the potential of SVMs for image classification. In order to study how different low-level features affect on the image classification accuracy, we first use the dominant color histogram. Unfortunately, the classification results could not satisfy our expectation (confusion matrix shown table 2). Combined features based on dominant color histogram and gabor-based textures can give a very high classification accuracy for the training samples (matrix shown in table3). For simplicity, we labels eight different classes as following: brick (A), grass (B), leaves(C), pebbles (D), road (E), sky (F) and tree (G).

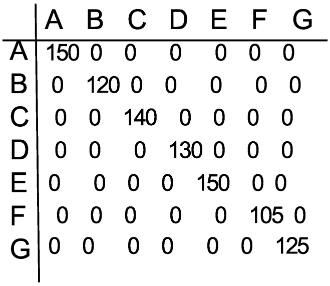
From the above results, we will conduct the following experiments based on the combined the features. In order to study the different kernels provided in table 1 on image classification, different kernel parameters and regularization parameter C are needed to be adjusted to achieve the better generalization. The most straightforward method to select the parameters is to use validation data, which are distinct from the training data. However, when the amount of available training data is limited, it is important to have an alternative means of tuning the parameters, without having to put aside parts of the training set for validation purposes. The proposed method for tuning the parameters is mainly based on Scholopf's work [19]. The strength of an SVM is based on its automatic capacity control (in terms of VC-dimension). The capacity control takes place within a class of functions $\{f_\alpha : \alpha \in A\}$ (or equivalently F) specified a prior by its choice of the kernel function. Scholopf pointed out the VC-dimension of f_α can be estimated by term that is proportion to the radius r of the smallest ball containing all the data points in F. Since the r depends on the shape of F, the VC-dimension depends on the kernel parameters. The estimation of r from the given parameters can be formulated in a similar way to training SVMs. At the same time, we also use the different multi-class strategies on image classification. Figure3 and Figure4 give the classification accuracy results while applying different parameters and different multi-class strategies.

Table 2. Confusion Matrix based on dominant color



	A	B	C	D	E	F	G
A	50	0	35	0	25	5	35
B	0	100	10	5	5	0	0
C	0	5	120	0	8	0	7
D	5	9	21	10	50	14	21
E	0	10	5	5	75	45	10
F	0	0	0	0	5	100	0
G	0	0	55	10	5	0	55

Table 3. Confusion Matrix based on combined feaures



	A	B	C	D	E	F	G
A	150	0	0	0	0	0	0
B	0	120	0	0	0	0	0
C	0	0	140	0	0	0	0
D	0	0	0	130	0	0	0
E	0	0	0	0	150	0	0
F	0	0	0	0	0	105	0
G	0	0	0	0	0	0	125

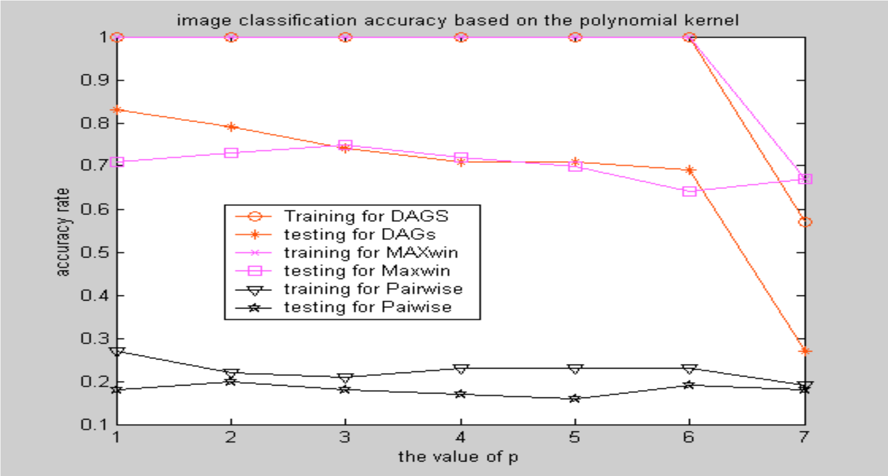


Fig. 3. Image classification result based on polynomial kernel (C=100)

From the results, we can see that the performance of DAGs is almost as the same as that of Maxwin strategy. However they performed better than Pairwise coupling. In the polynomial kernel method, with increasing of parameter p, the accuracy began to decrease. And when $p > 6$, the image accuracy became so bad. In the Gaussian kernel method, with decrease of the parameter α, the image accuracy began to increase. For specified problems, we should select the optimal parameters to make sure of the good generalization of the SVMs.

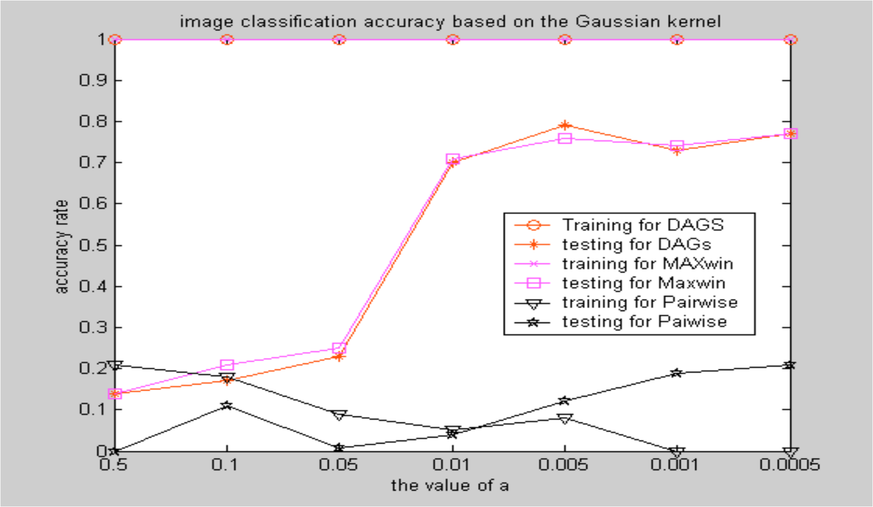


Fig. 4. Image classification accuracy based on Gaussian kernel (C=100)

Table 4. Comparison between SVMs and Neural network

Classifier	Average accuracy
Neural Network	50~70%
DAGSVM	60~80%

Compared the results applying neural networks [12], the provided maxwin and DAGs SVM receives a litter better performance (see table 4). In contrast to neural network, an SVM is more concerned with the minimizing the testing error than directly minimizing the training error.

6 Conclusion

In this paper, we only consider one image only consists of one object. However, images may consist of more objects. Therefore, how to classify such images becomes the goal of our further research.

References

1. D.A.Forsyth, J.Malick, M.M.Fleck, etc. “Finding pictures of objects in large collections of images,” in International Workshop on Object recognition for computer vision, (Cambridge, England), April 13-14 1996.
2. H-H.Yu and W.Wolf, ”Scene classification methods for image and video databases.” In Proc. SPIE, Digital Image Storage and archiving Systems, pp. 363-371,1995
3. M.M.Gorlcani and R.W.Picard, “Texture orientation for sorting photos “ at a glance”, in 12th Intl conference on Pattern Recognition, pp,459-464, October,1994
4. Oliver chapelle, Patrick Haffner and Vladimir N.Vapnik “support vector machines for histogram-based image classification”, IEEE Trans on neural networks, vol, 10. No.5 September,1999.
5. Kwang In Kim, Keechul Jung, Se Hyun Park and Hang Joon Kim, “support vector machine for texture classification”, IEEE Trans on pattern analysis and machine intelligence, vol, 24,NO.11, November 2002
6. V.Vpanik, *The Nature of Statistical Learning Theory, New York*: Springer Verlag, 1995.
7. Yan ni Wang, Long bin Chen and Bao gang Hu, “Semantic extraction of the building images using support vector machines”, Proceedings of First International Conference on Machine Learning and Cybernetics, pp 1608-1613, November 2002
8. DaShan Gao ,Jie Zhou and Leping Xia, “svm-based detection of moving vehicles for automatic traffic monitoring”, IEEE Intelligent Transportation Systems Conference Proceeding Oakand (CA) ,pp745-749,August,2001
9. B.Scholkof, K.Sung, C.J.C.Burges, F.Girosi, T.Pogio, and V.Vapnik, “comparing support vector machines with Gaussian Kernels to Radial Basis Fuction Classifiers.” IEEE ,Trans. Signal Processing, vol,45,no.11 pp2758-2765,1997
10. Vailaya, A. K. Jain and H.-J. Zhang, “On image classification: city images and landscapes”, Pattern Recognition, vol. 31, pp 1921-1936, December, 1998
11. M. Szummer and R. Picard, "Indoor-Outdoor Image Classification," IEEE International Workshop on Content-Based Access of Image and Video Databases CAIVD '98, Bombay, India, Jan. 1998
12. M. Markou, M. Singh and S. Singh, “Neural Network Analysis of Minerva Scene Analysis Benchmark”, 11th International Conference on Image Analysis and Processing, Palermo, Italy (26-28 September, 2001).
13. J. Platt ,”Fast training of support vector machines using sequential minimal optimization” in Advances in Kernel Methods --- Support Vector Learning, B. Scholkopf, C. Burges, and A. Smola, eds., pp. 185--208, MIT Press, 1999.
14. T.M.Cover, ”Geometrical and Statistical Properties of systems of Linear Inequalities with applications in pattern recognition”. IEEE Trans. Electronic Computers, vol.14, pp326-334, 1965.
15. J.Mercer, “Functions of positive and negative type and Their connection with the theory of integral equations”, Trans. London Philosophical Soc.(A),vol,209,pp415-446,1909
16. J.H.Friedman. Another approach to polychotomous classification, Technical report, Standford University, Department of Statistics, 10:1895-1924,1998
17. Trevir Hastie, RobertTibshirani,”classification by pairwise coupling”, Technical report, Stanford Univ and Univ of Toronto,1996, in proceeding of NIPS
18. .S.Manjunath, W. Y. Ma.” Texture features for browsing and retrieval of image data” IEEE Transactions on Pattern Analysis and Machine Intelligence,18(8),837-842,1996.
19. B.schollkopf,C.burges and V.Vapnik “Extracting support Data for Given Task” Proc.Int’l Conf. Knowledge Discovery and Data mining,pp252-257,1995

Machine Learning Approaches for Inducing Student Models

Oriana Licchelli, Teresa M.A. Basile, Nicola Di Mauro, Floriana Esposito,
Giovanni Semeraro, and Stefano Ferilli

Dipartimento di Informatica - Università degli Studi di Bari
via E. Orabona, 4 – 70125 Bari - Italia
{licchelli, basile, nicodimauro, esposito, semeraro, ferilli}@di.uniba.it

Abstract. The main issue in e-learning is student modelling, i.e. the analysis of a student's behaviour and prediction of his/her future behaviour and learning performance. Indeed, it is difficult to monitor the students' learning behaviours. A solution is the exploitation of automatic tools for the generation and discovery of user profiles, to obtain a simple student model based on his/her learning performance and communication preferences, that in turn allows to create a personalized education environment. This paper focuses on Machine Learning approaches for inducing student profiles, respectively based on Inductive Logic Programming (the INTHELEX system) and on methods using numeric algorithms (the Profile Extractor system), to be exploited in this environment. Moreover, an experimental session has been carried out, comparing the effectiveness of these methods along with an evaluation of their efficiency in order to decide how to best exploit them in the induction of student profiles.

1 Introduction and Motivations

In all areas of the e-era, personalization plays an important role. Specifically, in e-learning a main issue is student modelling since it is not easy to monitor students' learning behaviours. Adaptive personalized e-learning systems could accelerate the learning process by revealing the strengths and weaknesses of each student. They could dynamically plan lessons and personalize the communication and didactic strategy.

Artificial Intelligence (AI) offers powerful methods, which are useful in the development of adaptive systems. In the past, several intelligent techniques have been experimented in the ITS (Intelligent Tutoring Systems) development: in particular, AI techniques were exploited for the representation of pedagogical knowledge, the construction of the knowledge bases related both to the subject domain and to the didactic strategies and, finally, the student model generation, based on explicit knowledge of the student behaviour or on the analysis of the student mistakes and misunderstandings. Using AI techniques, Computer-Assisted Instruction systems can be adapted, during the interaction, to the student personality, characteristics and learning performance.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 935-944, 2004.

However, still today, many teaching systems based on the Web have not capitalized such experience and are often not capable to personalize the instruction material they supply in order to satisfy the needs of each single student. Anyway, a lot of attention has been given to user modelling in e-learning systems: for instance, EUROHELP [2] was devised to provide tools and methods for developing Intelligent Help Systems; InterBook [3] provided a user model based on *stereotype*, which represented the user's knowledge levels about every domain concept, and was modified as the user moved through the information space. Other projects used specific criteria to define a user ability model (e.g. MANIC [12], an online courseware system, determines user typology through heuristics, such as which slides the student has seen and which quizzes he/she has done).

In this paper we have focused our attention on two different Machine Learning approaches in order to discover the user-student preferences, needs and interests and to generate simple student models based on the learning performance and the communication preferences. In particular, we have used two different systems for inducing student profiles, respectively INTHELEX (based on Inductive Logic Programming) and the Profile Extractor system (that exploits algorithms based on numeric methods).

Assuming to have a first set of students and to be able to group them in a number of classes, each of which represents a student category, it is possible, by means of inductive methods of Machine Learning, to infer the concepts, i.e. the intensional definitions of student classes, which represent the student models. The training set from which to infer the conceptual user-student models (profiles) is made up of data concerning each student. Such data were initially collected through preliminary tests to estimate the students' background knowledge and to gather information concerning their educational goals and motivations, the preferred modalities of communication etc. Then, they were enriched by the logs of the successive interactions.

After illustrating both systems, we will provide a comparison of the effectiveness of these methods along with an evaluation of their efficiency in such an environment, in order to better understand how they could be exploited in an e-learning platform.

2 The Profile Extractor System

The Profile Extractor System, developed at LACAM (Knowledge Acquisition and Machine Learning Laboratory of the University of Bari) is a system to generate user profiles automatically [1]. It is a highly reusable module that allows the classification of users through the analysis of their past interaction with the system and employs supervised learning techniques.

Figure 1 shows the complete system architecture, that is further subdivided into four modules: Profile Rules Extractor, Profile Manager, Usage Patterns Extractor and XML I/O Wrapper.

The Profile Manager and the Profile Rules Extractor are the modules mainly involved in the profile generation process; the Usage Patterns Extractor groups dialogue sessions in order to infer some usage patterns that can be exploited for understanding user trends and for grouping single users, who share the same interests and preferences, into user communities [8]. The XML I/O Wrapper is the layer responsible for the integration of the inner modules with external data sources (using the XML protocol) and for the extraction of the data required for the learning process.

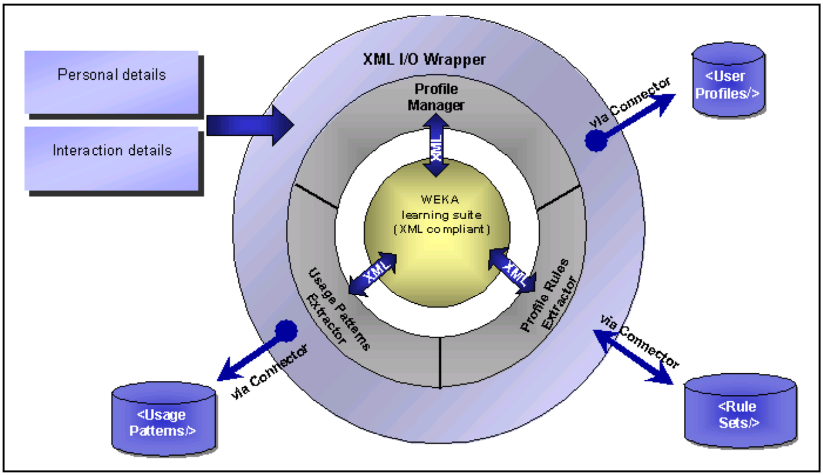


Fig. 1. The architecture of the Profile Extractor

The input to the Profile Extractor is represented by the XML file that contains the personal and interaction data of the user. This information is arranged into a set of unclassified instances, where each instance represents a single user, from the XML I/O Wrapper. The subset of the instances chosen to train the learning system has to be preclassified by a domain expert (each user is associated with a subset of the categories): this is the actual input to the Profile Rules Extractor, which will infer classification rule sets. The actual user profile generation process is performed by the Profile Manager, on the grounds of the user data and the set of rules induced by the Profile Rules Extractor. When the need to generate/update user profiles arises, the user data are arranged into a set of instances which represents the input to the Profile Manager. On the basis of the classification rule sets inferred, the classifier predicts the user behaviour in a system.

For the purpose of extracting user profiles, we focused on supervised machine learning techniques. Starting from preclassified examples of some target concepts, these techniques induce rules useful for predicting the classification of further unclassified examples. For this reason the core of the Profile Extractor is WEKA [13], a machine learning tool developed at the University of Waikato (New Zealand), that provides a uniform interface to many learning algorithms, along with methods for pre/post-processing and for the evaluation of the results of learning schemes, when applied to any given dataset. To integrate WEKA in the Profile Extractor we developed XWEKA, an XML compliant version of WEKA, that is able to represent input and output in XML format. The learning algorithm adopted in the profile generation process is based on PART [5], a rule-based learner that produces rules from pruned partial decision trees, built using C4.5's heuristics [9]. The antecedent, or precondition, of a rule is a series of tests, just like the tests at nodes in the classification path of a decision tree, while the consequent, or conclusion, gives the class that applies to instances covered by that rule. The main advantage of this method is not performance but simplicity: it produces good rule sets without any need for global optimization.

Extensive experimentation of the system proposed for the automatic extraction of the user profile has been carried out in a field not far from that of e-learning: digital libraries. We experimented the Profile Extractor System in digital libraries in several contexts like e-Commerce [1] and contemporary European cultural documents [6].

3 The Symbolic Learning System: INTHELEX

INTHELEX[1] (INcremental THEory Learner from EXamples) [4] is a symbolic learning system for the induction of *hierarchical* first-order logic theories from positive and negative examples. It can focus the search for definitions by exploiting the *Object Identity* bias (according to which terms denoted by different names within a formula must refer to different objects) [11]. Such a system is able to learn simultaneously *multiple concepts* (i.e., definitions for different concepts), possibly related to each other; it guarantees validity of the theories on all the processed examples; it uses feedback on performance to activate the theory revision phase; in addition to the possibility of refining a previously generated version of the theory, learning can also start from an empty theory. It exploits a previous version of the theory (if any), a graph describing the dependence relationships among concepts, and an historical memory of all the past examples that led to the current theory.

Incremental learning is necessary when either incomplete information is available at the time of initial theory generation, or the nature of the concepts evolves dynamically, which are unnegligible issues for the task of learning student profiles. Indeed, the classical batch models, that perform learning in one step and hence require the whole set of observations to be available from the beginning, are not able to handle such cases, that are very frequent in real-world situations. Thus, the need for incremental models to complete and support the batch ones.

The learning cycle performed by INTHELEX can be described as follows. A set of examples of the concepts to be learned, possibly selected by an expert, is provided by the environment. This set can be subdivided into three subsets, namely training, tuning, and test examples, according to the way in which examples are exploited during the learning process. Specifically, training examples, previously classified by the expert, are stored in the base of processed examples, and are exploited to obtain a theory that is able to explain them. Such an initial theory can also be provided by the expert, or even be empty. Subsequently, the validity of the theory against new available examples, also stored in the example base, is checked by taking the set of inductive hypotheses and a tuning/test example as input and producing a decision that is compared to the correct one. In the case of incorrectness on a tuning example, the cause of the wrong decision can be located and the proper kind of correction chosen, firing the theory revision process. In this way, tuning examples are exploited

[1] INTHELEX is currently available in binary format for i586 DOS-based platforms (`http://lacam.di.uniba.it:8000/systems/inthelex/`).

incrementally to modify incorrect (too weak or too strong) hypotheses according to a data-driven strategy. Test examples are exploited just to check the predictive capabilities of the theory, intended as the behavior of the theory on new observations, without causing a refinement of the theory in the case of incorrectness.

Another peculiarity of INTHELEX is the integration of multistrategy operators that may help to solve the theory revision problem by pre-processing the incoming information [7]. The purpose of induction is to infer, from a certain number of significant observations, regularities and laws that may be valid for the whole population. INTHELEX incorporates two inductive refinement operators, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative examples. Exceptions are exploited as a last resort when no correct refinement can be found. Deduction is exploited to fill observations with information that is not explicitly stated, but is implicit in their description, and hence refers to the possibility of better representing the examples and, consequently, the inferred theories. Indeed, since the system is able to handle a hierarchy of concepts, some combinations of predicates might identify higher level concepts that are worth adding to the descriptions in order to raise their semantic level. The concepts hierarchy contains all the dependencies among the concepts to be learned; if there are no expressed relations, the system will assume the concepts to be isolated. Such relations are expressed as a set of clauses like the following:

bicycle(X) :- wheel(X), mechanic(X,Y)
mechanic(X,Y):-bicycle_chain(X,Y,Z),front_gear(X,Y),rear_gear(X,Y)

whose interpretation is, respectively: "concept *bicycle* depends on concepts *wheel* and *mechanic*"; and "concept *mechanic* depends on concepts *bicycle_chain*, *front_gear* and *rear_gear*". In the graph variables are used as placeholders to indicate the concepts arity.

INTHELEX exploits deduction to recognize such concepts and explicitly add them to the example description. The system can be provided with a Background Knowledge containing complete or partial definitions expressed as the theory rules. Abduction aims at completing possibly partial information in the examples, adding more details. Its role in INTHELEX is helping to manage situations where not only the set of all observations is partially known, but each observation could also be incomplete. Abducibles are the predicates on which assumptions (abductions) can be made; integrity constraints provide indirect information on them. The proof procedure implemented in INTHELEX corresponds, intuitively, to the standard Logic Programming derivation suitably extended in order to consider abducibles and integrity constraints. Lastly, Abstraction removes superfluous details from the description of both the examples and the theory. The exploitation of abstraction in INTHELEX concerns the shift from the language in which the theory is described to a higher level one. An abstraction theory contains information on the operators according to which the shift is to be performed. INTHELEX automatically applies it to the learning problem at hand before processing the examples. The implemented abstraction operators allow the system to replace a number of components with a compound object, to decrease the granularity of a set of values, to ignore whole objects or just part of their features, and to neglect the number of occurrences of a certain kind of object.

4 Experimental Sessions

CAMPUS ONE is an e-Learning Project of the University of Bari, (`http://www.campusone.uniba.it`), for providing courses on Fundamentals of Computer Science for all types of degree (human degree, science degree, etc.). In this project, each student for each kind of degree must attend the first two modules (*Module 1 Fundamentals Computer Science, Module 2 Management Computer And File*). An experiment was performed for each module, by identifying three classes for each of them on the ground of the final student performance evaluation: good, sufficient or insufficient. The information on each student were gathered from the log file of an e-learning platform.

4.1 Design of the Experiments

The experimental dataset was made up of information on 140 students. The students'profiles were classified, by a domain expert, as *Good*, *Sufficient*, or *Insufficient* in the two modules above mentioned. The distribution of students into the three classes for both the considered modules are as follows. For module 1: 45% Good – 23,57% Sufficient – 31,43% Insufficient; for module 2: 45% – 21,43% – 33,57% respectively.

```
class_good(user_id) :-
   average_duration_acces(user_id,19),
   curriculum_name(user_id,ecdlprevenzioneinambientedilav),
   job_description(user_id,tecnico_della_prevenzione),
   number_access(user_id,41),
   initial_score_mod1sec1(user_id,25),
   final_score_mod1_sec1(user_id,100),
   initial_score_mod1_sec2(user_id,30),
   final_score_mod1_sec2(user_id,81),
   initial_score_mod2_sec1(user_id,35),
   final_score_mod2_sec1(user_id,48),
   initial_score_mod2_sec2(user_id,15),
   final_score_mod2_sec2(user_id,92),
   initial_score_mod3_sec1(user_id,45),
   final_score_mod3_sec1(user_id,56),
   initial_score_mod3_sec2(user_id,12),
   final_score_mod3_sec2(user_id,65), ...
```

Fig. 2. Example of student's description

The descriptions of the students, exploited as examples to induce classification rules, were made up of personal data of the students, such as background culture and current job (e.g., *curriculum_name* and *job_description*), and enriched by logs of their successive interactions with the e-learning platform, such as initial and final score that the student obtained for each section in each module, or number of accesses and duration of each access. Figure 2 shows a part of the example description for a student.

The goodness of profiles induced by the two approaches was evaluated according to a 10-fold cross-validation methodology, that is the 90% of the whole dataset was used for the training process and the remaining 10% for the testing phase. This process was made ten different times assuring that there was no intersections between the ten test sets. Several metrics were used in the testing phase and classification effectiveness has been measured in terms of the classical Information Retrieval notions – Precision (Pr) and Recall (Re) – and predictive accuracy [10].

More in detail, let the classes be $\{d_1 = \text{Good}, d_2 = \text{Sufficient}, d_3 = \text{Insufficient}\}$, for each value d_i, TP (True Positive) is the number of test users that both the system and the domain expert assigned to class d_i in the selected experiment. TN (True Negative) is the number of users that both the system and the domain expert did not classify as d_i. FP (False Positive) is the number of test users that the system classified as d_i in the selected experiment, differently from the domain expert classification (not d_i) in the same experiment. FN (False Negative) is the number of users that the system did not classify as d_i while the domain expert classified them as d_i.

Then, Recall, Precision and Accuracy are computed as follows:

$$\text{Re} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{Pr} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TOT}}$$

where TOT indicates the total number of test users.
We also used F-measure, which is a combination of Precision and Recall:

$$\text{F} = \frac{2 \times \text{Re} \times \text{Pr}}{\text{Pr} + \text{Re}}$$

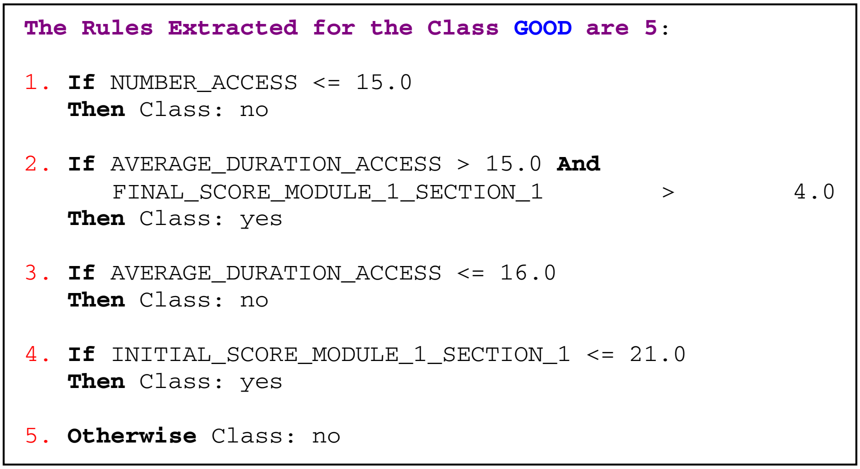
4.2 Discussion

For each class (Good, Sufficient and Insufficient), the systems were trained to infer proper classification rules, on the basis of an instance set representing different students (training set).

Figure 3 shows the classification rules describing the Good class of the Profile Extractor system (where the rule sets may be expressed as disjunctions of conditions) for the experiment set up on the first module, *Module 1 Fundamentals of Computer Science*, on the ground of logs containing interaction and student features, while Figure 4 shows an example of rule induced by INTHELEX for the same experiment.

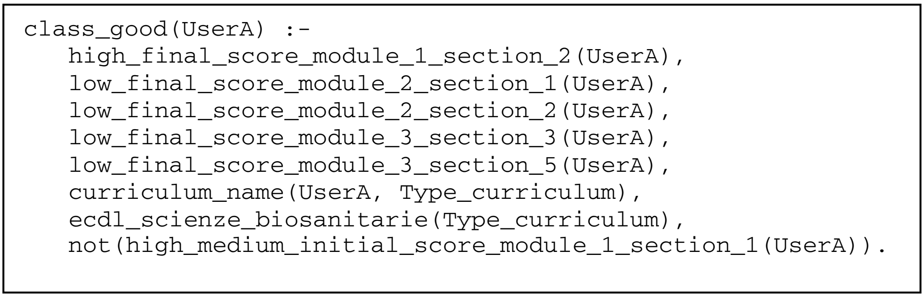
The rule learned by INTHELEX for the class Good, shown in Figure 4, says, among other things, that a *UserA* is a good student for module 1 if he didn't obtain an high result (score) at the beginning of the first section (section 1) of such a module, but he has filled his gap before the end of the module; in fact, he has obtained a high result (score) in the final test of the last section (section 2) of such a module.

The rule learned by INTHELEX for the class Good, shown in Figure 4, says, among other things, that a *UserA* is a good student for module 1 if he didn't obtain an high result (score) at the beginning of the first section (section 1) of such a module, but he has filled his gap before the end of the module; in fact, he has obtained a high result (score) in the final test of the last section (section 2) of such a module.



```
The Rules Extracted for the Class GOOD are 5:

1. If NUMBER_ACCESS <= 15.0
   Then Class: no

2. If AVERAGE_DURATION_ACCESS > 15.0 And
      FINAL_SCORE_MODULE_1_SECTION_1        >        4.0
   Then Class: yes

3. If AVERAGE_DURATION_ACCESS <= 16.0
   Then Class: no

4. If INITIAL_SCORE_MODULE_1_SECTION_1 <= 21.0
   Then Class: yes

5. Otherwise Class: no
```

Fig. 3. An example of classification rules returned by the Profile Extractor system



```
class_good(UserA) :-
   high_final_score_module_1_section_2(UserA),
   low_final_score_module_2_section_1(UserA),
   low_final_score_module_2_section_2(UserA),
   low_final_score_module_3_section_3(UserA),
   low_final_score_module_3_section_5(UserA),
   curriculum_name(UserA, Type_curriculum),
   ecdl_scienze_biosanitarie(Type_curriculum),
   not(high_medium_initial_score_module_1_section_1(UserA)).
```

Fig. 4. An example of classification rules returned by the INTHELEX system

Table 1 reports the experimental results concerning the classification effectiveness for both the experiments: *Module 1 Fundamentals Computer Science* and *Module 2 Management Computer And File*.

Both the experiments concerned the two modules that the students must attend. As shown in Table 1, the performance of the Profile Extractor system is greater than that of INTHELEX for all the evaluated metrics. An in-deep analysis of such a result suggests that the motivation of this behaviour is located in the inborn nature of the two systems: the Profile Extractor was built to *extract profiles* from a set of data that have an attribute-value representation, whereas the INTHELEX learning capabilities are more suitable in discovering relationships among objects in the representation. Therefore, since the dataset representing the student features is a set of attribute-value couples, the performance of the Profile Extractor system is more effective. Nevertheless, while the rules induced by the Profile Extractor give poor information about student data, the rules discovered by INTHELEX provide a lot of details both on the student's personal information and on their learning capability.

Table 1. Results for the "*Module1*" experiment.

			Re	**Pr**	**F-mea**	**Accuracy**
Module 1	**Good**	**Profile Extractor**	0,97	0,97	0,96	0,97
		INTHELEX	0.48	0.60	0.51	0.62
	Sufficient	**Profile Extractor**	0,97	0,97	0,96	0,97
		INTHELEX	0.35	0.54	0.41	0.57
	Insufficient	**Profile Extractor**	0,91	0,92	0,90	0,94
		INTHELEX	0.61	0.53	0.52	0.69
Module 2	**Good**	**Profile Extractor**	0,97	0,92	0,94	0,95
		INTHELEX	0.39	0.52	0.42	0.54
	Sufficient	**Profile Extractor**	0,97	0,96	0,96	0,98
		INTHELEX	0.43	0.61	0.41	0.78
	Insufficient	**Profile Extractor**	0,88	0,90	0,88	0,91
		INTHELEX	0.67	0.64	0.60	0.74

Thus, a conclusion of this analysis is that a combination of both approaches could be effectively applied in the e-learning environment in order to adapt the platform, during the interaction, to the student personality, characteristics and learning performances. In fact, a co-operation of both methods could consist in, firstly, discover a profile with an high accuracy, by means of the Profile Extractor, in order to perform a phase of adaptation of the platform to a coarser grain-size level, and successively to fine-tune the personalization of the platform exploiting the more detailed profiles induced by INTHELEX.

5 Conclusion and Future Work

E-learning environments give users a high degree of freedom in following a preferred educational path, together with a control to explore effective paths. This freedom and control is beneficial for the students, resulting in a deeper understanding of the instructional material. Sometimes, this type of e-learning environment is problematic, since some students are not able to explore it effectively. One way to address this problem is to augment the environments with personalized support. Indeed, it is possible to adapt an e-learning environment planning a personalized path for each user-student, based on his needs, goals and characteristics, with the aim of improving the learning process.

Paper focused on student modelling, and presented two systems for automatically generating the profiles of an e-learning user. We have evaluated the effectiveness of both systems in such an environment and we have observed that they seem to be complementary for improving the personalization of the e-learning platform.
Future work concerns the exploitation of the induced profiles in order to plan a personalized educational path.

Acknowledgement. The authors would like to thank Professor T. Roselli for the CAMPUS ONE dataset used in this experimentation.

References

1. Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Semeraro, G., Zambetta, F.: Improving the usability of an e-commerce web site through personalization. In: Ricci, F., Smyth, B. (Eds.): Recommendation and Personalization in Ecommerce. Proc. RPeC'02, Malaga, Spain (2002) 20-29
2. 2 Breuker, J. editor: EUROHELP: Developing Intelligent Help Systems. Conceptual model of intelligent help system. EC, Copenhagen (1990) 41-67
3. Brusilovsky, P., Eklund, J.: A Study of User Model Based Link Annotation. Educational Hypermedia. Journal of Universal Computer Science 4, 4 (1998) 429-448
4. Esposito, F., Ferilli, S., Fanizzi, N., Basile, T.M.A., Di Mauro, N.: Incremental Multistrategy Learning for Document Processing. Applied Artificial Intelligence Journal, 17(8/9), Taylor & Francis, London (2003) 859-883
5. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. Proc. of the 15th International Conference on Machine Learning. Morgan Kaufmann (1998) 144-151
6. Licchelli, O., Lops, P., Semeraro, G., Bordoni, L., Poggi, F.: Learning preferences of users accessing digital libraries. In: Cha, J., Jardim-Gonçalves, R., Steiger-Garção, A. (Eds.): Concurrent Engineering – Advanced design, production and management systems. Proceedings CE'03, Madeira, Portugal (2003) 457-465
7. Michalski, R. S.: Inferential Theory of Learning. Developing Foundations for Multistrategy Learning. In: Michalski, R. S. Tecuci, G. (Eds.): Machine Learning. A Multistrategy Approach, volume IV, Morgan Kaufmann, San Mateo, CA. 3–61
8. Paliouras, G., Papatheodorou, C., Karakaletsis, V., Spyropoulos, C., Malaveta, V.: Learning User Communities for Improving the Service of Information Providers. LNCS 1513, Springer (1998) 367-384
9. Quinlan, J.R. : C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
10. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34 (1), (2002) 1–47
11. Semeraro, G., Esposito, F., Malerba, D., Fanizzi, N., Ferilli, S.: A logic framework for the incremental inductive synthesis of datalog theories. In: Fuchs, N. E. (Ed.): Logic Program Synthesis and Transformation, LNCS 1463, Springer-Verlag (1998) 300-321
12. Stern, M., Woolf, B.P., Kurose, J.F.: Intelligence on the Web? Proc. of the 8th World Conference of the AIED Society, Kobe, Giappone (1997).
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (2000).

Monte Carlo Approach for Switching State-Space Models

Cristina Popescu and Yau Shu Wong

Department of Mathematical and Statistical Sciences
University of Alberta, Edmonton, Alberta, T6G 2G1, Canada
popescu@math.ualberta.ca

Abstract. In this paper we present a Monte Carlo EM algorithm for learning the parameters of a state-space model with a Markov switching. Since the expectations in the E step are intractable, we consider an implementation based on the Gibbs sample. The rate of convergence is improved using a nesting algorithm and Rao-Blackwellised forms. We illustrate the performance of the proposed method for simulated and experimental physiological data.

1 Introduction

In recent years, various switching state-space models (SSMs) were developed with applications ranging from econometrics [1] to control engineering [2]. In this paper, we are mainly concerned with the models introduced in [3] and represented mathematically by

$$X_t^{(m)} = A^{(m)}X_{t-1}^{(m)} + w_t^{(m)}, \quad w_t^{(m)} \sim N(0, Q^{(m)}), \quad m = 1, \ldots M, \tag{1}$$

$$Y_t = C^{(S_t)}X_t^{(S_t)} + v_{t,} \quad v_t \sim N(0, R), \tag{2}$$

where $\{Y_t\}$ is the sequence of observations, $\{X_t^{\{m\}}\}$ are the M sequences of real valued hidden state vectors, and $\{S_t\}$ is the sequence of discrete hidden state vectors. The discrete switching state can take M values, $S_t \in \{1,\ldots,M\}$, and has a homogeneous Markovian structure specified by the initial probabilities π and the transition matrix Φ. The zero-mean Gaussian noise vectors $w_t^{(m)}$, and v_t are uncorrelated with covariance matrices $Q^{(m)}$, m=1,…, M, and R, respectively, and they are independent of the sequence $\{S_t\}$. $A^{(m)}$ are the transition matrices and $C^{(m)}$ are the output matrices for the state-space model m.

Notice that for M=1 (i.e. no discrete switching variable), we obtain a linear Gaussian state-space model. The Expectation Maximization (EM) algorithm, with the E step based on the Kalman filter and smoother, can be applied for system identification [4]. On the other hand, if we keep only the observation sequence $\{Y_t\}$ and the discrete switching variables $\{S_t\}$, we are in the classical setting of hidden Markov chains. For learning the parameters, we can employ a special case of the EM algorithm, known as the Baum-Welch algorithm ([5] pp 329-333).

The E step becomes intractable [3], if we try to implement the EM algorithm for the switching SSM (1)-(2). To overcome this difficulty, two approaches using ap-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 945-954, 2004.

proximate inference methods were employed (see [6] for a justification of the convergence of these variants of the EM algorithm). The first method merges a mixture of M Gaussian into one Gaussian at each time step ([1], pp. 99-109). The second one is based on a variational approach [3]. Both approaches suffer the well-known problems of the EM algorithm, namely the possibility to be trapped in a local minimum, and the slow rate of convergence

In this paper, we apply a Monte Carlo EM (MCEM) algorithm [7] and we approximate the expectations in the E step using a block Gibbs sampler [8]. In a purely Bayesian approach, the Gibbs sampler was previously used for a slightly different model (see [1], pp.237-241). In our case, the advantage of using the Gibbs sampler is that, once the simulated values for the discrete variables $\{S_t\}$ are obtained, we can apply the Kalman filter and smoother. This allows us to use Rao-Blakwellised forms in the E step, and also to speed up the algorithm by nesting [9]. Once the parameters of the system are estimated, we can solve a classification problem, assigning each observation Y_t to the class i, $i\in\{1,\ldots,M\}$ corresponding to the maximal probability $P(S_t=j|\{Y_t\})$, j=1,…,M. The proposed nested Monte Carlo EM (NMCEM) algorithm is presented in detail in the next section.

Being a combination between discrete and linear Gaussian dynamics, the switching SSMs are appropriate for modeling many real life problems. Here, we first compare the performance of our approach with the two methods mentioned earlier, for simulated data. Then, we consider a set of physiological experimental data, which is divided into training and test set. On the training set, we apply the NMCEM for learning the parameters of the models. On the test set, the estimated values of the parameters are used for classification. The results are reported in section 3.

2 The Learning Algorithm

The parameters of the proposed model are $\Theta=\{R, \Phi, \pi, A^{(m)}, C^{(m)}, Q^{(m)}, \Xi^{(m)}, \mu^{(m)}, m=1,\ldots,M\}$. Here, $\Xi^{(m)}$ and $\mu^{(m)}$ are the covariance and the mean of the hidden Gaussian state variable $X_1^{(m)}$ in the state-space model m.

2.1 The Monte-Carlo EM Algorithm

The EM algorithm is especially suitable for learning the parameters of the switching SSM (1)-(2), because it is easier to calculate the likelihood of the augmented data $\{S_t, X_t^{(m)}, m\in\{1,\ldots,M\}, Y_t,\}$ than the likelihood of the observed data $\{Y_t\}$ (see [3] for detailed calculations). The algorithm starts with an initial guess Θ_0 of the unknown parameters and then iteratively compute the estimation Θ^*, each iteration involving two steps: the expectation (E) and the maximization (M) step.

In the E-step, the conditional expectation $Q(\Theta,\Theta_n)$ of the augmented data log-likelihood is computed, giving the current estimation Θ_n of the parameters. For the mixture structure of the switching SSMs, any attempt to calculate the exact expectation becomes impractical [3]. Here, we propose an MCEM algorithm [7]. Instead of an exact E-step, we use an approximation

$$Q(\Theta, \Theta_n) \approx \frac{1}{L_n} \sum_{l=1}^{L_n} \log P\left(\left\{ S_t(l), X_t^{(m)}(l), m = 1 \ldots M, Y_t \right\} \middle| \Theta \right), \tag{3}$$

where $\left\{ S_t(l), X_t^{(m)}(l), m = 1 \ldots M \right\}$, l=1,…,L$_n$ is a sample from $P\left(\left\{S_t, X_t^{(m)}, m = 1 \ldots M\right\} \middle| \{Y_t\}, \Theta_n\right)$.

The M-step performs the maximization with respect to the parameters Θ. In our case, we maximize $Q(\Theta, \Theta_n)$ by setting

$$A^{(m)}(n+1) = \sum_{t=2}^{T} E\left[X_t^{(m)} X_{t-1}^{\prime(m)} \middle| Y, \Theta_n \right] \left(\sum_{t=2}^{T} E\left[X_{t-1}^{(m)} X_{t-1}^{\prime(m)} \middle| Y, \Theta_n \right] \right)^{-1}, \tag{4}$$

$$Q^{(m)}(n+1) = \frac{\sum_{t=2}^{T} \left(E\left[X_t^{(m)} X_t^{\prime(m)} \middle| Y, \Theta_n \right] - A^{(m)}(n+1) E\left[X_{t-1}^{(m)} X_t^{\prime(m)} \middle| Y, \Theta_n \right] \right)}{T-1}, \tag{5}$$

$$C^{(m)}(n+1) = \sum_{t=1}^{T} Y_t E\left[1_{\{m\}}(S_t) X_t^{(m)} \middle| Y, \Theta_n \right] \left(\sum_{t=1}^{T} E\left[1_{\{m\}}(S_t) X_t^{(m)} X_t^{\prime(m)} \middle| Y, \Theta_n \right] \right)^{-1}, \tag{6}$$

$$R(n+1) = \frac{1}{T} \left(\sum_{t=1}^{T} Y_t Y_t' - \sum_{m=1}^{M} C^{(m)}(n+1) \sum_{t=1}^{T} E\left[1_{\{m\}}(S_t) X_t^{(m)} \middle| Y, \Theta_n \right] Y_t' \right), \tag{7}$$

$$\mu^{(m)}(n+1) = E\left[X_1^{(m)} \middle| Y, \Theta_n \right], \quad \Xi^{(m)}(n+1) = \mathrm{VAR}\left(X_1^{(m)} \middle| Y, \Theta_n \right), \tag{8}$$

$$\pi^{(m)}(n+1) = P\left(S_1 = m \middle| Y, \Theta_n \right) \tag{9}$$

$$\Phi^{(m,q)}(n+1) = \frac{\sum_{t=2}^{T} P\left(S_t = m, S_{t-1} = q \middle| Y, \Theta_n \right)}{\sum_{t=1}^{T-1} P\left(S_t = q \middle| Y, \Theta_n \right)}, \tag{10}$$

where T is the number of observations, Y:= $\{ Y_t \}$, for any set Ω, 1_Ω denotes the characteristic function of Ω, and VAR(X) denotes the variance of X. The expectations in (4)-(10) are approximated in the Monte-Carlo E step using the Gibbs sampler.

2.2 The Gibbs Sampler

Let denote X:=$\{ X_t^{(m)}, m = 1 \ldots M \}$, S:=$\{ S_t \}$ and define

$$f(S \mid X, Y) := P(S_T \mid X, Y)\prod_{t=1}^{T-1} P(S_t \mid S_{t+1}, X, Y), \tag{11}$$

$$g(X \mid S, Y) := \prod_{m=1}^{M} P(X_T^{(m)} \mid S, Y)\prod_{t=1}^{T-1} P(X_t^{(m)} \mid X_{t+1}^{(m)}, S, Y), \tag{12}$$

$$f(S \mid Y) := P(S \mid Y),\ g(X \mid Y) := P(X \mid Y). \tag{13}$$

The probabilities in (11) and (12) can be calculated as in [8] (notice that in (12) we have only Gaussian densities). The probabilities in (13) are in fact intractable, since they involve integration with respect to X and S, respectively. Starting with an initial guess S(0), the block Gibbs sampler generates a sequence X(1), S(1), X(2), S(2), … such that S(i) is drawn from f(S|X(i),Y) and X(i+1) is drawn from g(X|S(i),Y). This produces a homogeneous Markov chain {S(i),X(i)} and, under appropriate regularity conditions ([10]), we are eventually sampling from P(S,X|Y). In our case, strong convergence results can be easily proved if {S(i)} is aperiodic and irreducible [10] (for instance if $\Phi^{(i,j)}>0$ for any i,j=1,…,M). We can now state the following theorem

Theorem 1. If {S(i)} is aperiodic and irreducible, then {X(i)} is an ergodic Markov chain with invariant distribution g(X|Y). Moreover, The Markov chain {X(i)} is ϕ-mixing.

Proof. The proof is similar to the proof of theorem 1 reported in [11]. The main idea is to transfer some of the properties of {S(i)} to {X(i)} using a duality principle ([11]).

Corollary 1 We have the following central limit theorem

$$\frac{1}{\sqrt{I}}\sum_{i=1}^{I}\left(h(X(i)) - E^g\left[\,|h(X)|\,|Y\right]\right) \to N(0, \sigma_h) \tag{14}$$

for any function h such that $E^g[\,|h(X)|^2\,|Y]<\infty$ and with

$$0 < \sigma_h = VAR^g(h(X)) + 2\sum_{i=1}^{\infty} COV^g(h(X(0)), h(X(i))) < \infty \tag{15}$$

Proof**.** This is a direct consequence of the geometric ϕ-mixing property of {X(i)}.

Remark 1. A similar result can be formulated for the Markov chain {S(i),X(i)} with the stationary distribution P(S, X | Y)=g(X | S, Y)f(S | Y)

Remark 2. A very important consequence of the theorem is the fact that the ergodic theorem [10] applies and we can approximate

$$E^g\left[h(X) \mid Y\right] \approx \frac{1}{I}\sum_{i=1}^{I} h(X(i)) \tag{16}$$

for I large enough and $E^g[\ |h(X)|\ |\ Y\ ]<\infty$. In our case, this justifies the Rao-Blackwellised forms. The central limit theorem in the corollary allows us to monitor the convergence by estimating the variance σ_h.

2.3 The Nested MCEM Algorithm

Since under very reasonable conditions the proposed Gibbs sampler is convergent, we can then use the following Rao-Blackwellised forms and the Kalman smoothing for estimating the expectations in (4)-(8):

$$E\left[\,1_{\{m\}}(S_t)X_t^{(m)}\ \middle|\ Y\,\right] \approx \frac{\sum_{i=1}^{L_n} E\left[\,1_{\{m\}}(S_t(i)\,)X_t^{(m)}\ \middle|\ S(i)\,,\,Y\,\right]}{L_n}, \tag{17}$$

$$E\left[\,1_{\{m\}}(S_t)X_t^{(m)}X_t^{\prime(m)}\ \middle|\ Y\,\right] \approx \frac{\sum_{i=1}^{L_n} E\left[\,1_{\{m\}}(S_t(i)\,)X_t^{(m)}X_t^{\prime(m)}\ \middle|\ S(i)\,,\,Y\,\right]}{L_n}, \tag{18}$$

$$E\left[\,X_t^{(m)}X_t^{\prime(m)}\ \middle|\ Y\,\right] \approx \frac{\sum_{i=1}^{L_n} E\left[\,X_t^{(m)}X_t^{\prime(m)}\ \middle|\ S(i)\,,\,Y\,\right]}{L_n}, \tag{19}$$

$$E\left[\,X_t^{(m)}X_{t-1}^{\prime(m)}\ \middle|\ Y\,\right] \approx \frac{\sum_{i=1}^{L_n} E\left[\,X_t^{(m)}X_{t-1}^{\prime(m)}\ \middle|\ S(i)\,,\,Y\,\right]}{L_n}. \tag{20}$$

We also have

$$P(\,S_t\ =\ m\ |\ Y)\ \approx\ \frac{\sum_{i=1}^{L_n} 1_{\{m\}}(S_t(i)\,)}{L_n}, \tag{21}$$

$$P(\,S_t\ =\ m,\,S_{t-1}\ =\ q\ |\ Y)\ \approx\ \frac{\sum_{i=2}^{L_n} 1_{\{m\}}(S_t(i)\,)1_{\{q\}}(S_{t-1}(i)\,)}{L_n}. \tag{22}$$

In every cycle of the MCEM, we repeatedly apply random generators. Hence, compared to any deterministic implementation of the EM algorithm, the MCEM is slow. Thus, any technique which is capable to improve the rate of convergence is very valuable. Here we propose a nested MCEM algorithm [9].

We consider two augmented-data sets $Y_{aug1}=\{Y_t,\ S_t\}$ and $Y_{aug2}=\{\ Y_t,\ S_t,\ X_t^{(m)},\ m=1,\dots,M\}$. The NMCEM algorithm fixes the augmented data Y_{aug1} and runs several EM iterations conditional on these values. The E-step for the inner EM algorithm is based only on the Kalman smoother and the previous Rao-Blackwellised forms and, as a consequence, is more efficient. A formal proof of the faster rate of convergence for the nested EM is given in [9]. Here, we present the algorithm in detail:

```
Initialize the parameters of the model
Repeat until log likelihood has converged:
   Draw values for $\{S_t, X_t^{(m)}, m=1,\dots,M\}$ using the
   Gibbs sampler;
   Calculate the conditional probabilities using for-
   mulas (21)-(22);
   For k=1 to $K_{nesting}$ do
     Inner E-step
     Run Kalman smoothing recursions given the sample
     $\{S(i), i=1,\dots,L_n\}$;
     Replace in the Rao-Blakwellised forms (17)-(20);
     Inner M-step
     Re-estimate the parameters of the model using the
     formulas (4)-(10).
```

We conclude this section by discussing some issues concerning the implementation of the algorithm. Following the suggestion given in [7], we increase the number L_n of data in the Gibbs sampler at each new cycle of the MCEM algorithm. For the number of inner steps we have tried $K_{nesting}$=3,5,7 with comparable results. Since the log likelihood is intractable, we apply a harmonic average [12] to estimate its values, and the bridge sampling presented in [9] to monitor the convergence. Instead of the classical block Gibbs sampler presented in [8], we prefer the disturbance smoother from [13], for its better performance.

3 Applications

Now, to assess the effectiveness of the proposed algorithm, we compare the performance of the MCEM with the algorithms presented in [3] and [1], pp 99-109. To facilitate the comparison with the variational approach, we consider similar data as in [3].

3.1 Simulated Data

First, we suppose that the model parameters are known and we use the Gibbs sampler to solve a classification problem. As in [3], the SSMs and the switching process are defined as

$$X_t^{(1)} = 0.99X_{t-1}^{(1)} + w_t^{(1)}, \quad w_t^{(1)} \sim N(0,1), \tag{23}$$

$$X_t^{(2)} = 0.9X_{t-1}^{(2)} + w_t^{(2)}, \quad w_t^{(2)} \sim N(0,10), \tag{24}$$

$$Y_t = X_t^{(S_t)} + v_{t,} \quad v_t \sim N(0,0.1), \tag{25}$$

where the initial probabilities are $\pi^{(1)}=\pi^{(2)}=0.5$, and the transition probabilities are $\Phi^{(1,1)}= \Phi^{(2,2)}=0.95$ and $\Phi^{(1,2)}= \Phi^{(2,1)}=0.05$.

We generate 200 sequences of length 200 from this model. A set of data and its true segmentation are displayed in Fig. 1.

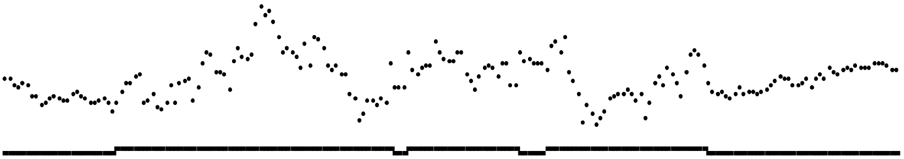


Fig. 1. Data sequence of length 200 with its true segmentation bellow it: the upper dots represent the switch state 2, and the lower dots the switch state 1

For each sequence we run the Gibbs sampler described in section 2.2. A point Y_t is considered to be from class 1 or class 2 according to the values of the probabilities $P(S_t=m \mid Y_1,\ldots,Y_T)$, m=1,2. These conditional probabilities are estimated using (21). The histogram of the percent correct segmentations is displayed in Fig.2(a). Compared with the results reported in [3], even with the annealed algorithm, we notice a significant increase of the percent correct segmentations. However, the Gibbs sampler is much slower than the variational approach, and we could not use exactly the same data as in [3].

For the same simulated data sets, we have also implemented the merging method presented in [1], pp. 99-106. Since we are not interested in the Kalman smoothing, we apply a generalized pseudo Bayesian algorithm of order 1 to each SSM. Using only the forward part of the algorithm, we classify according to $P(S_t=m| Y_1,\ldots,Y_t)$, m=1,2, and the results are displayed in Fig2(b). The average performance of the method based on the Gibbs sample is about 8.4% better than this Gaussian merging method. However, we can approximate the conditional probabilities $P(S_t=m| Y_1,\ldots,Y_T)$, m=1,2 using the merging method and the smoother presented in [1] pp. 106-109. The results are shown in Fig.2(c) and are obviously better than the ones displayed in Fig.2(b). The average of the percentage of the correct segmentations is about 3.8% greater for the algorithm based on the Gibbs sampler.

We also tested other switching SSMs, and we have consistently obtained better results with the algorithm based on the Gibbs sampler than with the algorithm based on Gaussian merging.

3.2 Experimental Data

We now consider the experimental physiological data illustrated in Fig.3(a). We apply the NMCEM for system identification. These data correspond to a patient tentatively diagnosticated with sleeping apnea[1]. We notice that the respiration pattern is charac-

[1] Data are available online at http://www.physionet.org/physiobank/database/santa-fe.

terized by a succession of no breathing, gasping breathing and normal rhythmic breathing (see also [3] and the references therein).

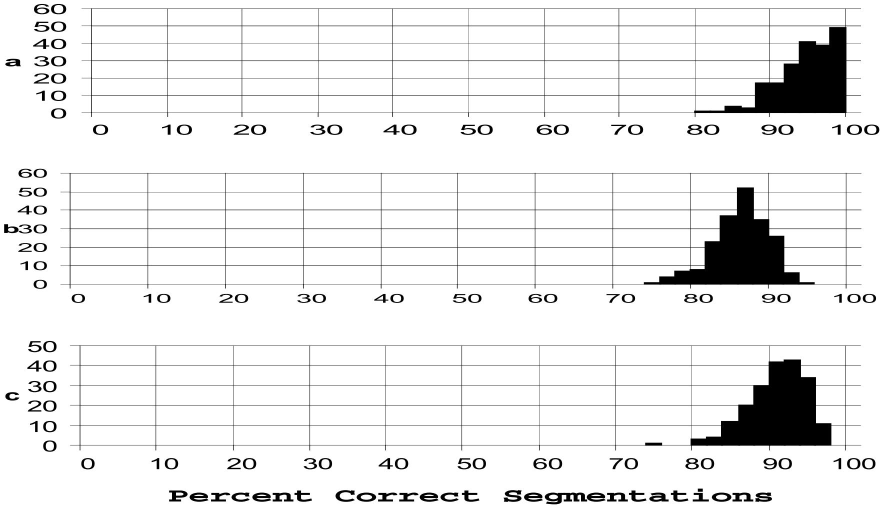


Fig. 2. Histogram of percent correct segmentations: (a) inference based on the Gibbs sample (b) Gaussian merging, forward algorithm (c) Gaussian merging, with approximated smoothing

In [3] it is shown that a switching SSM with M=2 components and the dimension of the state-space K=2 is the most suitable model for this data set. One component is specialized for the gasping and normal breathing and the other component models the data during periods of apnea. We have used the same type of switching SSM and we have applied the NMCEM for estimating the parameters. The results are very similar to those reported in [3]. The segmentation found is shown at the bottom of Fig.3(a)

As we have already mentioned the likelihood is intractable, and we can find only an estimation of the log likelihood for each cycle of the NMCEM. In Fig.4 we compare the MCEM with the NMCEM for K_{nested}=3. Using nesting we reduce the time at least by a factor of two. The final estimated value of the maximum likelihood is similar to the value of the lower bound obtained in [3].

The results for the test set are displayed in Fig.3(b). Using the MCEM or any of the nested versions, we obtain estimations for the maximum log likelihood lower than the value of the bound reported in [3] (our best values are around -0.75 nats per observation compared to -0.85 nats per observations reported in [3]). Also, the segmentation seems to be more accurate (see Fig.3(b)) since it is capable of detecting possible apnea periods undiscovered in [3]

4 Conclusions

In this paper, we present a nested Monte Carlo EM algorithm based on the Gibbs sampler for learning the parameters of a state space models with a Markovian switching. The proposed algorithm seems to be a good alternative when the execution time is not critical and we want to increase the accuracy of the results.

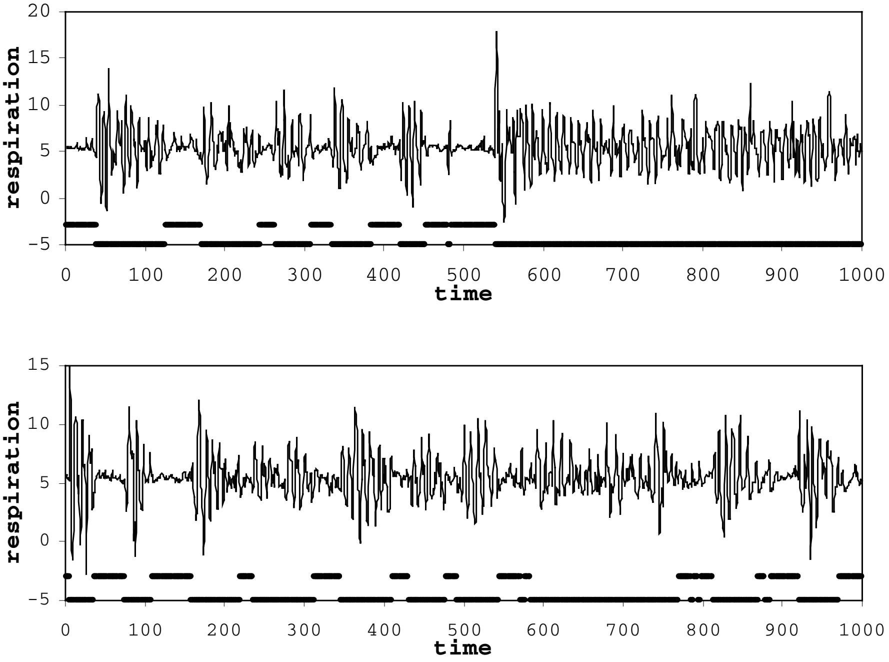


Fig. 3. Chest volume of a patient with sleep apnea. Measurements are sampled at 1 Hz. (a) Training data (b) Test set

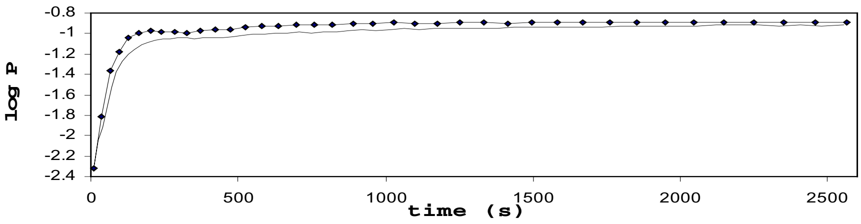


Fig. 4. Learning curves for a MCEM (plain line) and a NMCEM (dotted line)

We compare our approach to two existing methods, both based on the EM algorithm, but neither of them involving random number generators. Although our method is slower than the other two, the results reported here for the simulated and the experimental data are more accurate. We studied several sets of data corresponding to different switching state space models, and, when the parameter of the system are known, it seems that the proposed stochastic approach consistently provide a better classification. For system identification, more tests need to be done, since the performance of the EM algorithm is dependent on the initial guess and better results can only be a consequence of a more inspired initialization algorithm.

The Monte Carlo EM is a very time consuming method. Here we show that using nesting and Rao-Blackwellised forms, we can reduce the computing time by at least a factor of two. Moreover, the execution time of the nested Monte Carlo EM can be reduced even further using parallel computing.

The approach proposed in this paper can also be applied to other switching state space models. For example, it can be very easy adapted for the economical models studied in [1].

References

1. Kim, C.J, Nelson, C.R.: State Space Models with Regime Switching, The MIT Press (1999)
2. Shumway, R.H., Stoffer, D.S.: Dynamic Linear Models with Switching. J. Amer. Stat. Asoc. 86 (1991) 763-769
3. Ghahramani, Z., Hinton, G.E.: Variational Learning for Switching State-Space Models. Neural Computation vol.12 no. 4 (2000). 963-996
4. Shumway, R.H., Stoffer, D.S.: An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm. J. T Ser. Anal. vol. 3 no. 4 (1982) 253-264
5. McLachlan, G.J., Peel, D.: Finite Mixture Models, Wiley, New York (2000).
6. Neal, R.M., Hinton, G.E.: A New View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In: Jordan, M.I. (ed.): Learning in Graphical Models, Kluwer Academic Press, (1998)
7. Wei, G.C.G, Tanner, M.A.: A Monte Carlo Implementation of the EM Algorithm and the Poor Men's Data Augmentation Algorithm. J. Amer. Statist. Assoc. 85 (1990) 699-704.
8. Carter, C.K., Kohn, R.: On Gibbs Sampling for State Space Models. Biometrika, vol. 81 no 3 (1994) 541-553
9. Van Dyk, D. A.: Nesting EM Algorithms for Computational Efficiency. Statist. Sinica, 10 (2000) 203-225
10. Tierney, L.: Markov Chains for Exploring Posterior Distributions. Ann. Statist. Vol. 22 no.4 (1994), 1701-1728.
11. Diebolt, J., Robert, C.P.: Estimation of Finite Mixture Distributions through Bayesian Sampling. J.R. Statist. Soc. B, vol. 56 no. 2, (1994) 363-375.
12. Newton, M.A., Raftery, A.E.: Approximate Bayesian Inference with the Weighted Likelihood Bootstrap. J.R. Statist. Soc.B. vol. 56 no.1 (1994) 3-48.
13. Durbin, J., Koopman, S.J.: A Simple and Efficient Simulation Smoother for State-Space Time Series Analysis. Biometrika vol. 89 no.3 (2002), 603-615

Epistemological Remediation in Intelligent Tutoring Systems

Jos□phine Tch□tagni[1], Roger Nkambou[1], and Froduald Kabanza[2]

Department of Computer Science
[1] University of Quebec at Montreal
[2] University of Sherbrooke
tchetagn@info.uqam.ca, nkambou.roger@uqam.ca, kabanza@dmi.usherb.ca

Abstract. This paper presents an approach to student□s errors diagnosis in intelligent tutoring systems and to the remedial instruction to overcome those errors. Our contribution arises from two key components. Firstly, a diagnosis model, which based on the nature of the learner□s input, as well as its exercise knowledge model., uses Bayesian induction (*a posteriori maximization*) to find the most probable causes of a failure Secondly a remedial instruction model which will be the focus of this paper. This model will use the epistemological nature of the faulty skill that was diagnosed.

1 Introduction and Overview

Cognitive diagnosis and remedial instruction are fundamental elements of instruction. The importance of cognitive diagnosis is due to the fact that it guides the instructional plan: all the teaching actions depend on its result. According to Ohlsson (1987), cognitive diagnosis is the process which allows a tutor to assess the cognitive state of its pupil after any kind of performance has been observed. Direct remediation is an active subject in traditional education as in intelligent tutoring systems (ITS) (Woolf and Hall 1995). Remediation by means of instructional planning has also been widely studied (Wasson 1998). In this article, our goal is to propose a remediation framework at the finest level. Indeed, most models of instructional planning and remediation describe how a failure/error is detected, as well as the various approaches by which they could be overcome: presentation of similar examples or problems, analogies, simulation, explanations (Conati et al. 1997). These techniques proved their value in real educational contexts (Algebra-Tutor), but our idea is based on a different philosophy. Our goal is to provide formal knowledge models for remediation pedagogy. This mainly aims at providing ITS authors with computational models of remediation at the lowest level of this process: the level where interaction with the learner occurs. Using the domain of Logic Programming (LP) basics, we will show how this could be achieved if we assume that the whole process is centered on the epistemological features of the skills to remedy.

In some well-structured domains, the objective is to acquire first factual knowledge and propositions. Thereafter, the acquisition of procedural skills follows, and they may further be used in complex problem solving skills acquisition (Brien 1997). Our

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 955-966, 2004.

investigation relates to the diagnosis and remediation of students in ITS, which promote learning in that type of domain. We use the fact that the generic cognitive processes associated with each intellectual skill of the field can be modeled (Paquette 2002; Nkambou et al. 2003). In this way, our main assumption is that the internal unfolding of the remediation can be described in a more formal way. We will call *epistemological remediation* a remediation approach which uses the epistemological nature of a faulty skill.

The paper is structured as follows. Firstly, we will briefly present the architecture of a system we developed to support our propositions, as the presentation of those ideas will be illustrated with some examples from that system behaviour. Secondly, we describe how Gagne's epistemology (Gagne and Briggs 1993) can be instantiated in the LP context, stressing out the relevance of our perspective. Thirdly, the remediation framework is presented. Details on the diagnosis model may be found in Tchétagni and Nkambou 2004, it will not be presented here due to space limitations. Briefly, a part from the classical approaches to diagnosis models (buggy rules, constraint violation, etc.), we introduced another approach based on the Bayesian induction mechanism called *Most Probable Explanation*. This particular approach may be useful when the exhaustive problem space or task graph associated with a problem is not available or is not appropriate to represent the solving behaviour in that problem. Regarding remediation, we will emphasize on a clear distinction between two submodels: the knowledge model which refers to a particular remedial capability (it can be seen as a pedagogic model) and the knowledge model which refers to the processing needed to apply a remedial approach for a particular skill. Finally, we will discuss about the current research on the use of artificial intelligence tools to perform diagnosis and remediation in ITS, and conclude.

2 The Current System Architecture

The current system (figure 1) prototype consists of : 1) the learner model; 2) an profiler agent, which supports diagnosis and remediation functionalities; 3) learning exercises and problems. Each training exercise has an associated knowledge model. The exercise itself can be of various types depending on the reasoning process it demands. For this reason, 4 levels of exercise are differentiated: level 0 exercises related to the acquisition of concepts and propositions; level 1 exercises related to the acquisition of a rule via the capacity to apply or use it in the given context; level 2 exercises which deals with the application of a procedure through the use of suitable rules in the given context; level 3 exercises which relate to problem solving activities.

We used the CKTN (Nkambou et al. 2003) model to generate the LP course. These objectives are structured in a pre-requisite hierarchy. Each objective has learning activities. Each learning activity is either an instructional activity or a training activity. An instructional activity is supported by one presentation resource or more (text), we do not use this kind of resource as input to the student model. A training activity is supported by what we call intelligent resources, i.e. self-contained resources with their knowledge model or problem space and we shall see the rationale of this choice below.

The knowledge model associated with an exercise contains references to all the skills of the learner model that are used in that exercise. For levels 1 exercises and above, the knowledge model will have two components: 1) the default component, which contains references; 2) a dynamic component, which will either define the causality of each associated skills in the success of that exercise (what is the probability of success, given that one skill is acquired), or define a behavior model for that exercise whenever available (like a problem space, a procedural model, etc.). For procedures for example, the dynamic model will describe an execution sequence for the production rules to be used. For problem solving, the model will be a space problem. This approach is justified from the facts that: 1) the tutor should know what the student is doing at any time in order to adapt its strategies; 2) the learner model should be continuously updated while he is solving exercises and problems. The learner model overlays the objective hierarchy and is interpreted as a Bayesian network (Tchetagni and Nkambou 2002).

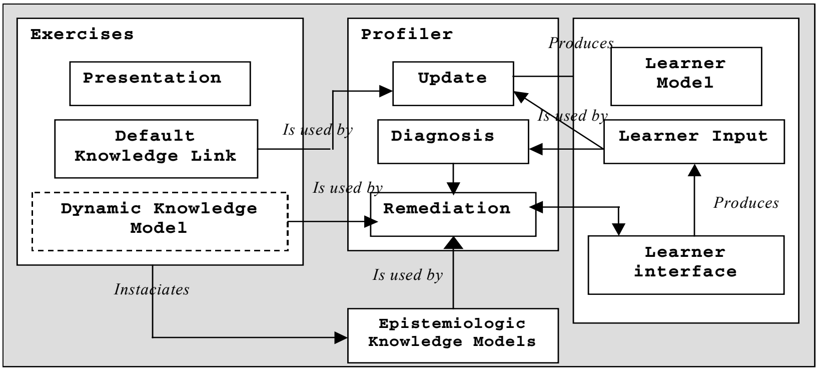


Fig. 1. System components and interactions

3 Knowledge and Skills in Logic Programming

A LP basic course can be conceived in 2 main modules: the language or grammar of terms acquisition and the application of the fundamental procedures of unification and resolution. The grammar of terms can be seen as a *concept* acquisition objective since all the terms in LP have attributes, which should be known in order to correctly identify terms. Unification of two compound terms may be seen as a two stages *procedure*, based on *rules* related to the unification of simple terms: first try to unify the 2 *functors* of each term, then try to unify *vis-à-vis* arguments in both terms. In the following, we use Gagn□ epistemology to describe formal epistemological models of skills, which will support remedial instruction. Thus, we consider skills such as: concepts, propositions, rules and procedures.

Concepts are objects sharing the same properties. Identification, classification, distinction are the abilities or cognitive processes most often associated with them in an LP course. Concept instance *identification* implies the recognition of its class attributes or basic characteristics. Figure 2 illustrates the corresponding model. Indeed, we tried to capture different philosophies of what is supposed to be a concept, in order to be able to manage a good range of situations. Thus as we can see, a concept is seen as a prototype with attributes as well as an entity defined by the relations between its properties or elements.

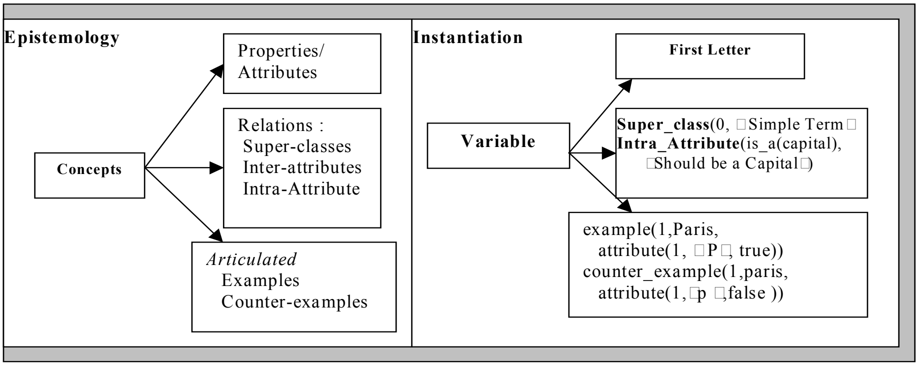

Epistemology
Concepts
Properties/ Attributes
Relations : Super-classes Inter-attributes Intra-Attribute
Articulated Examples Counter-examples
Instantiation
Variable
First Letter
Super_class(0, □Simple Term□ **Intra_Attribute**(is_a(capital), □Should be a Capital□)
example(1,Paris, attribute(1, □P□, true))
counter_example(1,paris, attribute(1,□p □,false))

Fig. 2. Epistemological models of concepts

In LP, rules are omnipresent. For example, the notion of dependent variable is built on a rule related to the state of a variable. The unification algorithm is based on applying rules, which themselves are built on the grammar of terms. It is thus of primary importance for diagnosis that the rules are correctly modelled in the system: their conditions and their consequences, an explicit expression to formulate the link between them (Figure 3). The conditions are propositions relating concepts while the consequences are other propositions or actions possibly performable by the learner. Then, procedures such as unification can easily be defined by specifying their phases, each one involving a set of rules. For the learner acquiring the ability to execute or apply a procedure, the most important thing is to know the ordering of the procedure stages. Then for each stage, the rule(s) that apply to the execution context should be applied or executed. Once again, remedial instruction will use these models.

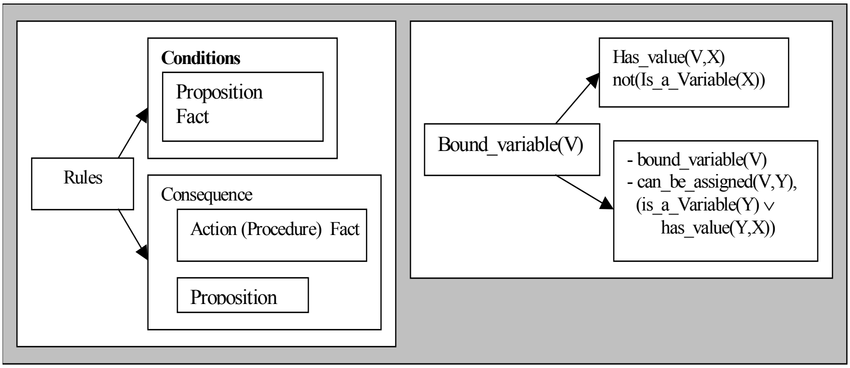


Fig. 3. Epistemological models for rules and procedures

3.1 Relevance of Epistemological Remedial Instruction

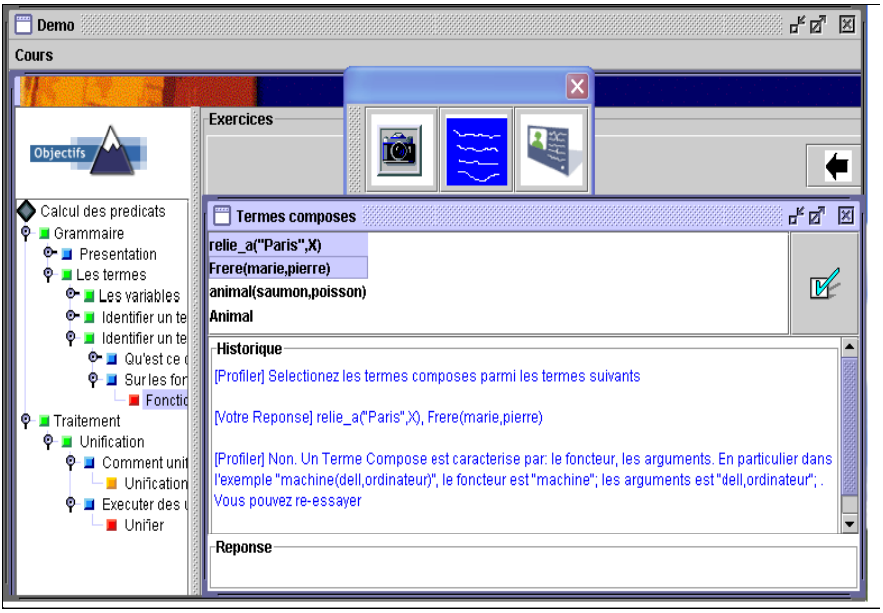
According to Lindsay and Norman 1997, acquiring a concept implies the possession of: that concept class, the associated sub-concepts (which are the attributes of the studied concept), the relationships between these sub-concepts, and if possible some examples of the concept. According to Brien 1997, concept acquisition is achieved by building schemas and these schemas correspond to its attributes with their inter-relationships. Therefore, learning is demonstrated by the ability to instantiate a schema for a particular concept instance. Our model of concept follows naturally this cognitive hypothesis. For production rules and principles, the condition and the action are built from propositions that use themselves some concepts. Acquiring the ability to apply or execute a procedure follows the same logic: it is necessary to include/understand the main process in terms of the sub-processes that make it up. Each sub-process generally corresponds to a set of rules. The sets of rules make it possible to determine the appropriate action to trigger in a given situation. Our procedural model is in agreement with these facts.

4 Remedial Instruction

Remediation is straightforward once the output of the diagnosis is given. Remediation should help the learner to understand the exercise and to solve it correctly after failure. We defined 2 remedial approaches: recall and articulation. In the next sections, we will outline how these approaches provide a more abstract model of remediation.

4.1 Recall

The learner’s model often reveals relevant information for the remediation process. For example, how to interpret the fact that a learner is unable to solve an exercise that is connected to a well mastered skill according to its model? Lack of attention or guessing is sometimes responsible for that (Siemers and Angelides 1998). Thus, rather than engaging an in-depth remediation, the system may proceed to a simple recall. While recall may be thought as a trivial remedial approach, its remedial aspect relies on its repetitive character, forcing the learner to *remind or recall an already experienced learned process or schema*. When the diagnosed problem is a concept, the system will state its attributes and most importantly *their values in the current problem context*, ensuring the transfer from theory to practice. This is where our “epistemological knowledge models” become important. They allow the tutor to dynamically generate this information. For a rule, the system will state its conditions and consequences again in the current context. Figure 4 shows an example of recall where the system detects that a pattern is missing in the solution, in this case, an animal (salmon, fish). Each item in the exercise is linked with an epistemological knowledge model instance (here the concept of compound term), which incorporates examples for concepts. Thus, the profiler analyzes this model, produces and enunciates its attributes and finally gives an example. A natural language converter is associated with the profiler. This process is very basic: it takes the propositions that comprise the epistemological knowledge model and produce a natural language version of them. For example, in the description: concept (variable,var), has a (co, first_letter), we state that the concept of □variable□has as a distinctive attribute called first letter. The converter can translate this kind of description into a human friendly phrase. Recall is similar to re-teaching. From an instructional and a learning point of views, recall is relevant either when the learner model informs that the diagnosed skill is supposed to be acquired, or when the learner□s solution is un-interpretable but incorrect. Recall also has a cognitive validity since according to Brien 1997, the acquisition of declarative knowledge may happen by: 1) subordination; 2) superposition; 3) composition. Recall relatively expresses the composition since it implies the combination of several known *schemas or frames* into a new frame that corresponds to the recalled element.



relie_a("Paris",X)
Frere(marie,pierre)
animal(saumon,poisson)
Animal

[Profiler] Selectionez les termes composes parmi les termes suivants

[Votre Reponse] relie_a("Paris",X), Frere(marie,pierre)

[Profiler] Non. Un Terme Compose est caracterise par: le foncteur, les arguments. En particulier dans l'exemple "machine(dell,ordinateur)", le foncteur est "machine"; les arguments est "dell,ordinateur"; . Vous pouvez re-essayer

Fig. 4. Knowledge model of recall processing and demonstration

4.2 Methodology of the Recall Remedial Instruction

The conduction of a remedial dialog depends on the corresponding remedial strategies. In the following, we explain how this is achieved when the recall is used. In this case, the tutor proceeds in 2 steps. First, he recalls the features of the diagnosed skill. Second, he prompts the student to answer again the same question (Figure 5). This pedagogical model is a template that shall be integrated in the overall pedagogical knowledge base of an intelligent tutor.

4.3 Articulation

Articulation-based remediation allows to come along with the learner in hulling the exercise in order to let him reach the solution *himself,* favouring in this way *knowledge construction*. In that sense, articulation is a significant way to remedy the student errors. Indeed, using a dialog, questions exploiting the nature of diagnosed skills are asked to the learner, in the context of the corresponding exercise. For example, if the *identification or recognition* of a concept is lacking, the system will ask the student to *enumerate the attributes* of this concept. If some attributes are absent or are attributed invalid values, the system will also question the student about those attributes

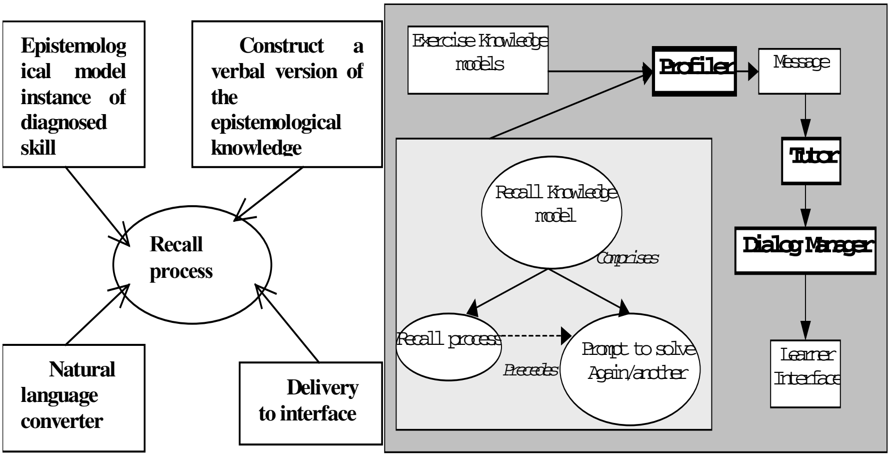
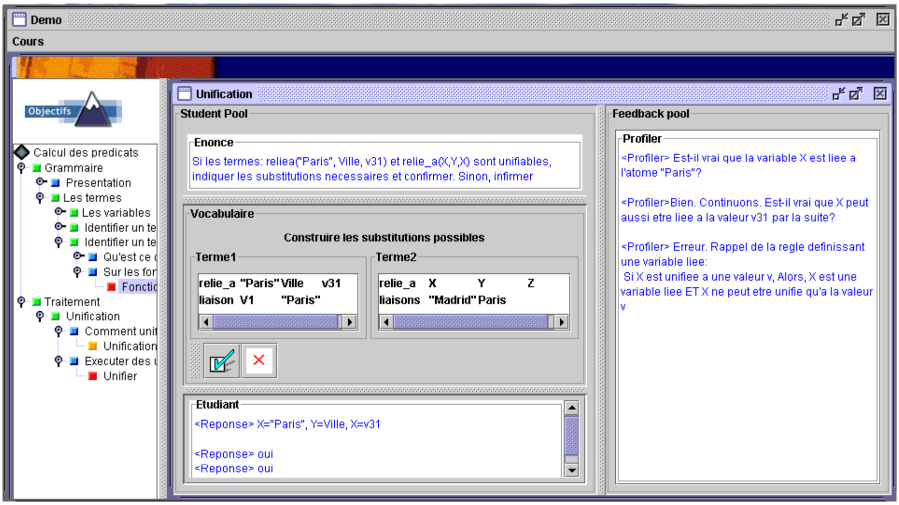

Epistemological model instance of diagnosed skill
Construct a verbal version of the epistemological knowledge
Recall process
Natural language converter
Delivery to interface
Exercise Knowledge models
Profiler
Message
Tutor
Dialog Manager
Learner Interface
Recall Knowledge model
Comprises
Recall process
Precedes
Prompt to solve Again/another

Fig. 5. Pedagogical knowledge model for the recall approach



Enonce

Si les termes: reliea("Paris", Ville, v31) et relie_a(X,Y,X) sont unifiables, indiquer les substitutions necessaires et confirmer. Sinon, infirmer

Terme1			
relie_a	"Paris"	Ville	v31
liaison	V1	"Paris"	

Terme2			
relie_a	X	Y	Z
liaisons	"Madrid"	Paris	

☑ ✕

Etudiant

<Reponse> X="Paris", Y=Ville, X=v31

<Reponse> oui
<Reponse> oui

Profiler

<Profiler> Est-il vrai que la variable X est liee a l'atome "Paris"?

<Profiler>Bien. Continuons. Est-il vrai que X peut aussi etre liee a la valeur v31 par la suite?

<Profiler> Erreur. Rappel de la regle definissant une variable liee:
Si X est unifiee a une valeur v, Alors, X est une variable liee ET X ne peut etre unifie qu'a la valeur v

Fig. 6. Making the learner articulate elements in concept and rule usage

Figure 6 illustrates a student□s error in solving a unification problem. He is unable to unify 2 terms while it is possible. The systems clearly diagnosed that the faulty rule is the one defining the notion of bound variable and its implications. Firstly, the profiler questions the learner on the conditions of this rule in the current problem context. Then the same is done on the consequences. If the learner□s difficulties persist, the systems enunciate the complete rule with an example. Note that even if one ends by stating the information on a skill as in a recall approach, this occurs at the end of a dialog with the learner, where he should probably have to realize the important aspects of the skill. The articulation of a procedure is based on the articulation of the production rules or principles, which correspond to each stage of that procedure. This is possible since the dynamic knowledge models of exercises that involve procedural and problem solving skills are based on production rules and principles.

Figure 7 and 8 illustrate the domain-dependant processing knowledge model associated with the articulation approach and the corresponding pedagogical knowledge model respectively. Briefly, the profiler uses the instantiated epistemological knowledge model associated with the diagnosed skill and uses its content to conduct articulation.

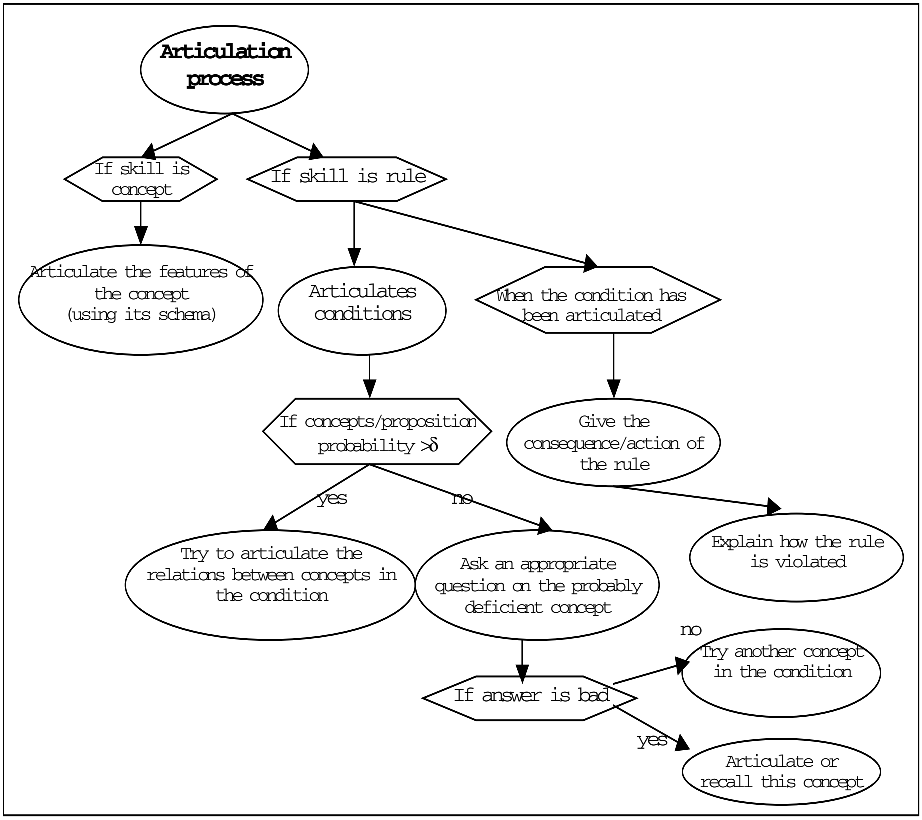

Articulation process
If skill is concept
If skill is rule
Articulate the features of the concept (using its schema)
Articulates conditions
When the condition has been articulated
If concepts/proposition probability >δ
Give the consequence/action of the rule
yes
no
Try to articulate the relations between concepts in the condition
Ask an appropriate question on the probably deficient concept
Explain how the rule is violated
no
Try another concept in the condition
If answer is bad
yes
Articulate or recall this concept

Fig. 7. Knowledge model of articulation processing

Articulation relates to knowledge construction or self-explanation. In this paper, we demonstrate a formal way to use this approach where the nature of the diagnosed skill is exploited as well as the cognitive processes underlying it. Our assumptions are also valid since Gagne et Briggs 1993 prescribe that concepts are better understood by pinpointing their distinctive characteristics. As well, pinpointing their components and the relationships between those components allows a better learning of rules and procedures.

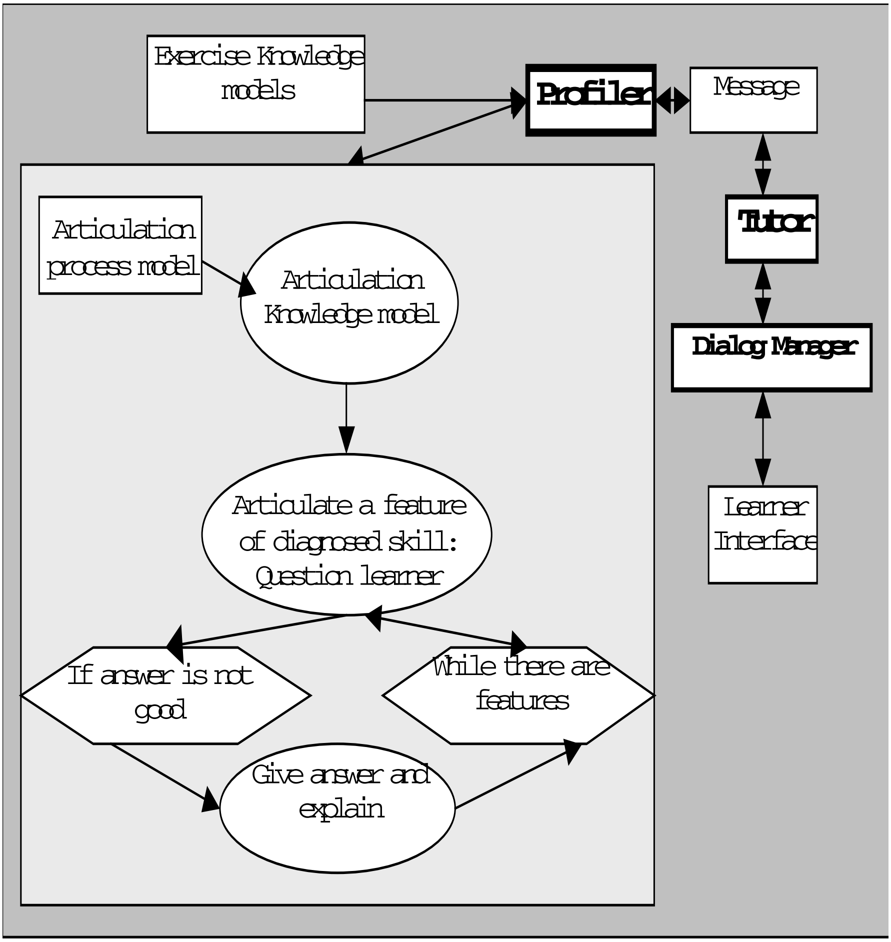

Exercise Knowledge models
Profiler
Message
Tutor
Dialog Manager
Learner Interface
Articulation process model
Articulation Knowledge model
Articulate a feature of diagnosed skill: Question learner
If answer is not good
While there are features
Give answer and explain

Fig. 8. Pedagogical knowledge model for remedial articulation

5 Related Work

Diagnosis and remediation is a hot issue in ITS research and almost all well known systems implement it. SOPHIE (Brown et al. 1982) performs model-based diagnosis but despite its natural language processing abilities, which allow immediate feedback, it is not able to constructively help the student understand its problem or reach the solution. Algebra Tutor (Koedinger et al. 1997) uses a model tracing technique to diagnose errors, to produce flags and suggestions when a mal-rule is encountered. Knowledge construction dialogs have been used by VanLehn et al. 2000 to help a learner encountering difficulties: the learner is presented with a new situation intended to provide him a deeper understanding of the current problem. Self-explanation (Conati and VanLehn 2000) is a tutorial approach where the learner is encouraged to self-explain their understanding of an example and its solution.

First of all, we think that the remedial instruction we propose here can be used by many of the preceding approaches. In fact, feedback, coaching and hinting consists in: 1) recalling a rule to the learner; 2) giving the appropriate rule when the tutor guesses what the learner is trying to do. The studies on natural language based dialog for knowledge construction is the approach most similar to articulation. Rather than providing the learner with the rule to apply in a given situation, the Atlas component of Van Lehn et al. decides when to trigger a dialog with the learner in order to make him understand the rule. Our articulation approach is similar. However, it extends to other knowledge elements such as concepts, and it uses the learner model. In fact, it could break down the articulation dialog when a rule is diagnosed and it happens that it is in fact a component of that rule (a concept) that is not well mastered.

6 Conclusion

Our main contribution in this paper is from a conceptual perspective. We proposed a formal framework integrating diagnosis and remediation approaches in *well-structured domains* such as LP. The proposed techniques have been successfully applied to exercises solving scenarios-simulation in a prototype basic course. We do not just recall or articulate the diagnosed faulty skills. Remediation also occurs in the context of the problem currently solved by a student. This was possible because each remedial approach was associated with 2 models: 1) a processing model which governs how our profiler-agent uses the contents of the epistemological knowledge model instances associated with a diagnosed skill to generate human friendly utterances; 2) a pedagogical model of remediation which dictates how the tutor should conduct the dialog with the learner in that particular remedial approach. Thus, our future work will contribute to the formalisation of pedagogical and content processing knowledge needed for epistemological remediation. Future work concerns the evaluation of this approach from the learner view: no matter if we present an example, a feedback, a hint, does the fact of exploiting the nature of knowledge in a remedial tutoring context enhance the learner performance?

Acknowledgements. This research is sponsored in part by the NSERC (National Sciences and Engineering center)-CANADA and Infotel.Inc., CANADA

References

Anderson, J. R., Boyle, C. F., and Reiser, B. J. 1985. Intelligent tutoring systems. *Science* 228: 456-462.

Brien, R. 1997. *Science cognitive et formation*. Presses de l'Universit□du Qu□bec, Sillery, Qu□bec.

Brown, J.,S., D. Burton and J. de Kleer,J. 1982. Pedagogical, natural language and engineering techniques in SOPHIE I, II and III. In *Intelligent Tutoring Systems*. Pp. 227-282. New York: Academic Press

Conati, C., Gertner, A.,S., VanLehn,K. and Druzdel, M.,J. 1997: Online Student Modeling for Coached Problem Solving Using Bayesian Network. In *Jameson, A.,* Pads, C., Tasso, C. (ed.): User Modeling - Proceedings of the Sixth Intemational Conference. Springer, New York.

Conati,C. and VanLehn,K. 2000. Further Results from the Evaluation of an Intelligent Computer Tutor to Coach Self-Explanation. In Proceedings of the 5th International Conference on Intelligent Tutoring Systems, 304-313. Berlin: Springer Verlag

Druzdzel, M. and Henrion, M. 1993. Efficient reasoning in qualitative probabilistic networks. In Proceedings of the 11th National Conference on Artificial Intelligence, 548-553 Menlo Park , Calif : AAAI/The MIT Press.

Gagne, R., Briggs, L. and Wager, W. 1993. Principles of instructional design. New York: Holt, Rinehart, and Winston.

Koedinger, K.,R., Anderson, J.R., Hadley, W.H., and Mark, M.A. 1997. Intelligent Tutoring Goes to School in the Big City. International Journal of Artificial Intelligence in Education, 8.

Lindsay P. H. and Norman D. A. 1997 *Human Information Processing: An Introduction to Psychology*, Academic Press, New York.

Nkambou,R., Gauthier,G. and Frasson,C. 2003 CREAM-Tools : An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems. In : Authoring Tools for Advanced Technology Learning Environments : Toward cost-effective adaptative, interactive, and intelligent educational software. Pp. 93-138. Kluwer Publishers.

Ohlsson, S. 1987 Some Principles of Intelligent Tutoring. In: Lawler, R. W. and Yazdani, M. (eds.) Artificial Intelligence and Education. 203 □238. Norwood, Ablex Pub.

Ohlsson, S. and Langley, P. 1986. Psychological evaluation of path hypotheses in cognitive diagnosis. In H. Mandl & A. Lesgold (Eds.), Learning issues for intelligent tutoring systems. New York: Springer

Paquette, G. 2002. Mod□lisation des connaissances et des comp□tences: un langage graphique pour concevoir et apprendre. Sainte-Foy:Les Presses de l□Universit□du Qu□bec.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Net- works of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA.

Siemers, S. and Angelides,S. 1998. Toward an intelligent tutoring system architecture that supports remedial tutoring. Artificial Intelligence 12:469-611.

Tchetagni,J. and Nkambou,R. 2002. Hierarchical Representation and Evaluation of the Student in an Intelligent Tutoring System. In Proceedings of the 6th International Conference on Intelligent Tutoring Systems, LNCS 2363, pp. 708-717. Berlin: Springer Verlag

Tchetagni,J. and Nkambou,R. 2004. A framework for epistemological diagnosis in Intelligent Tutoring Systems using artificial intelligence tools. CALIE□2004

VanLehn, K., Freedman, R., Jordan, P., Murray, C., Osan, R., Ringenberg, M., Rose, C., Schulze, K., Shelby, R., Treacy, D., Weinstein, A. and Wintersgill, M. (2000) Fading and deepening: The next steps for Andes and other model-tracing tutors. In Proceedings of the 5th International Conference onIntelligent Tutoring Systems, ITS 2000, edited by G. Gauthier, C. Frasson, and K. VanLehn. Berlin:Springer Verlag.

Wasson, B. 1998. Facilitating dynamic pedagogical decision making: PEPE and GTE. Instructional Sciences 26(3) 299-316

Wenger, E. (1987). Artificial Intelligence and Tutoring Systems. Morgan Kaufmann, Los Altos.

Woolf, B.,P. *and* Hall, W. 1995. Multimedia Pedagogues: Interactive Systems for Teaching and Learning. *IEEE Computer* 28(5):74-80

XML-Based Learning Scenario Representation and Presentation in the Adaptive E-learning Environment

P. Kazienko and J. Sobecki

Department of Information Systems, Wroclaw University of Technology
Wyb. Wyspianskiego 23, 50-370 Wroclaw, Poland
{kazienko,sobecki}@pwr.wroc.pl

Abstract. E-Learning environments nowadays should adapt to their users and computer platforms they are using. In this paper the learning scenario collaborative recommendation based on consensus method using XML is presented. We present the XML description of the learning scenario and its components as well as the link database and XSL transformations for presenting them on given platforms in the way suitable for particular students.

1 Introduction

The Information Society development requires that their members are able to utilize new ways of acquiring knowledge and new skills. Traditional learning environments are necessary for primary and secondary education, however for higher, postgraduate and other forms of additional education, different modern forms of education and training, such as e-Learning or Computer Based Teaching, are becoming vital nowadays. This is mainly because the times when knowledge acquired by people in the traditional education institutions such as schools or universities were sufficient for the whole life, have gone for ever.

Nowadays e-Learning environments based on the web technologies are attracting increasing number of very differentiated users. There are many reasons for such environments (especially Distant Learning) being so popular [11]: facilitating different retraining programs for participants from different areas; providing access to specialist centers; providing equal opportunities in any aspect; promotion prices for non-campus students; making courses more student directed; revitalizing less popular courses; linking exploded campuses; improving the quality of courses.

E-Learning environments have also some disadvantages, from which the almost complete reduction of personal contacts between students and teachers is of the greatest importance. The lack of these contacts hinder appropriate understanding between students and teachers what results in different problems in the education process, for which the problems of Tutoring Strategy (TS) selection are ones of the greatest significance.

The TS is defined as a combination of means and methods used in the whole didactic process to increase its effectiveness. The TS may be realized by the Learning Scenario (LS), which is the selection of the knowledge pieces as well as their

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 967-976, 2004.

sequence, connections and form of presentation. In many e-Learning environments despite the differences among students and their progresses only single TS is applied. In works [9, 14] the solution of this problem by means of application of collaborative LS selection, which offers adaptation of the LS according to other students' experiences, was presented together with its application. In this paper we presented the application of XML to represent most of the system elements, such as LS instances and student model but also to display the education material on the particular platforms. The XML proved to be suitable tool for representation and presentation of very complex data also in adaptive interface recommendation [15] and adaptive tutoring strategy selection [14] systems.

In the following section the XML-based strategy representation together with its components is presented. In the third section the XML-based representation of student model and the process of LS adaptation are given. The following section describes link extraction and utilization in learning scenario presentation. In the fifth section the method for the actual scenario extraction and presentation on the given platform are shown. The conclusions and future works perspectives are presented in the summary.

2 XML-Based Learning Scenario Representation

The different TS are usually designed to fit differences in students' learning styles, which in turn are consequences of differences in cognitive styles. According to the work [2] for each learning perspective one of the two contradictory values could be assigned: perception (sensory or intuitive), input (visual or auditory), organization (inductive or deductive), processing (active or reflective) and understanding (sequential or global). For each learning perspective corresponding teaching style could be assigned accordingly: content (concrete or abstract), presentation (visual or verbal), organization (inductive or deductive), student participation (active or passive) and perspective (sequential or global). Tutoring strategies are characterized with any combination of values assigned to the above mentioned perspectives. We could obtain a numerous set of different strategies if we want to meet all of them.

In the implementation of e-Learning environment for Polish traffic regulation (presented in [14]) only four main types of strategies were distinguished: textual, graphical, animated and active. It was obviously a kind of simplification of the above presented model. The actual tutoring strategy was built from elements of four different types, however only specified combinations of elements were supposed to be valid.

It is possible to design more flexible environments that are able to be used by very differentiated population of learners who are using various computer platforms. Let us consider the tutoring system which role is to present the material of some courses. We can divide the whole material into some parts, which could be called *concepts* that in turn are to be learned by a student. Depending on the learning strategy the sequence of concepts may be different. Also the concepts themselves could be presented in many different ways. For many traditional web-based e-Learning environments the concepts are presented by a sequence of hypermedia pages [8] - see fig.1. But when we consider not only differences among learners but also among computer platforms they are using, the model should not perceive the pages (for example HTML) as atoms.

In the proposed model we should also define the pages as separate elements, but on the pages we should distinguish other chunks of information that could be presented to the user, according to the user's and platform's requirements. The information presented on the pages could be in the following forms (media): text, image, sound, video or animation, and interactions (for example implemented in Macromedia Flash). The page definition contains also information about the content, its sequence of appearance and links to other pages.

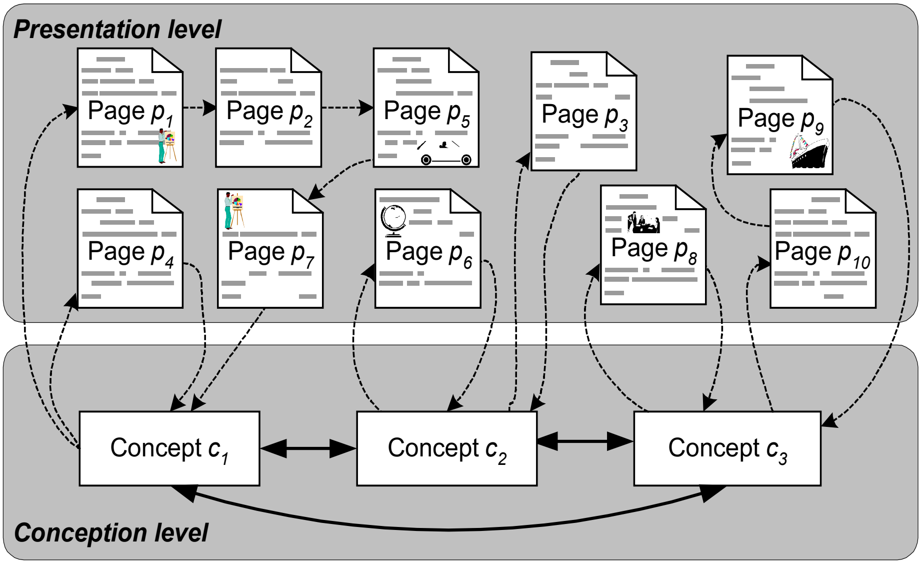


Fig. 1. Knowledge structure and its presentation.

The complete learning strategy contains not only concepts but also tests that verify the knowledge acquired by the student *st* belonging to the set of students *St* ($st \in ST$). The tests are also to control the learning process and help to decide whether repetitions or even changes in the learning strategy are necessary. So the learning scenario could be defined in the following way. Let *C* be the finite set of concepts to be learned and *T* be the finite set of tests checking the knowledge of all the concepts from the set *C*. The learning scenario s_{st} for a given student *st* (belonging to the set of scenarios $s_{st} \in S$) is defined as any reasonable order in *C*, where for each concept $c \in C$ there is assigned corresponding test $t_c \in T$. This, however, defines only *the conception level* of the learning scenario. To define *the content level* we must introduce the notion of pages, which are the content containers. Let *P* denote the set of pages, where each $p \in P$ contains corresponding resources in form of various media: texts, images, tables, audio, animation, video, interactive elements and hypermedia links. The content of each $c \in C$ is defined as reasonable ordered subset of pages $O \subset P$. All the resources have also information assigned, which define the computer platforms (*F*) they could be displayed or played on.

The XML representation of the learning scenarios could be defined as follows. First we shall define the XML document containing all the information about the atom elements (see the fragment of the XML file `media_el.xml` below):

```
<content_element element_id="e0021" type="video" storage="/video/001.avi">
     <platforms>
              <platform platform_id="p01">pc</platform>
              <platform platform_id="p04">kiosk</platform>
     </platforms>
 </content_element>
 <content_element element_id="e0274" type="text" storage="/texts/010.txt">
     <platforms>
              <platform platform_id="p01">pc</platform>
              <platform platform_id="p04">kiosk</platform>
              <platform platform_id="p03">pda</platform>
              <platform platform_id="p02">mobile</platform>
       </platforms>
 </content_element>
```

Then we are able to define the sample page (see bellow). The concepts and the learning scenarios could be defined respectively:

```
<page page_id="p121">
     <element element_id="e0102"/>
     <element element_id="e0133"/>
     <element element_id="e2109"/>
</page>
```

3 Learning Scenario Adaptation

In works [9,14] the method for consensus based learning scenario adaptation is presented in details. It is based generally on the ideas of collaborative recommendation [12]. The overall adaptation architecture is presented in fig. 2. The adaptation is based on the consensus methods, which are seeking for such solution that is the most representative for some subset of the whole set of elements, called also the profile [13].

We assume that in the e-Learning environment the population of students are learning one particular course. For each student the appropriate learning scenario in recommended. The recommendation in the first phase of the system life is based only on the expert knowledge (based on the stereotype reasoning), however with the growth of the number of learners the system is collecting its experiences that are the basis of the consensus based learning scenario adaptation. The recommendation of the scenario is made upon the consensus of the successful learning scenarios of the group of similar learners. To determine the groups, we introduce the notion of the learner profile and also the distance function between pairs of the learner profiles. Then using for example popular k-means clustering algorithm the learners are grouped by their profiles.

According to [7] the user model in the adaptive system consist of the user data and the usage data. The user data characterizes the user of the system and contains his or her demographic data (name, sex, address, occupation, education, customer data, psychographic data, etc.), user knowledge, user skills and user interests. The usage data contains information that may be observed directly from the user's interaction with web-based system [6]. In the e-Learning environment the user data may also contain information about learner cognitive characteristic [2] and the usage data contains the learners' history of learning scenarios already passed together with information of corresponding test results.

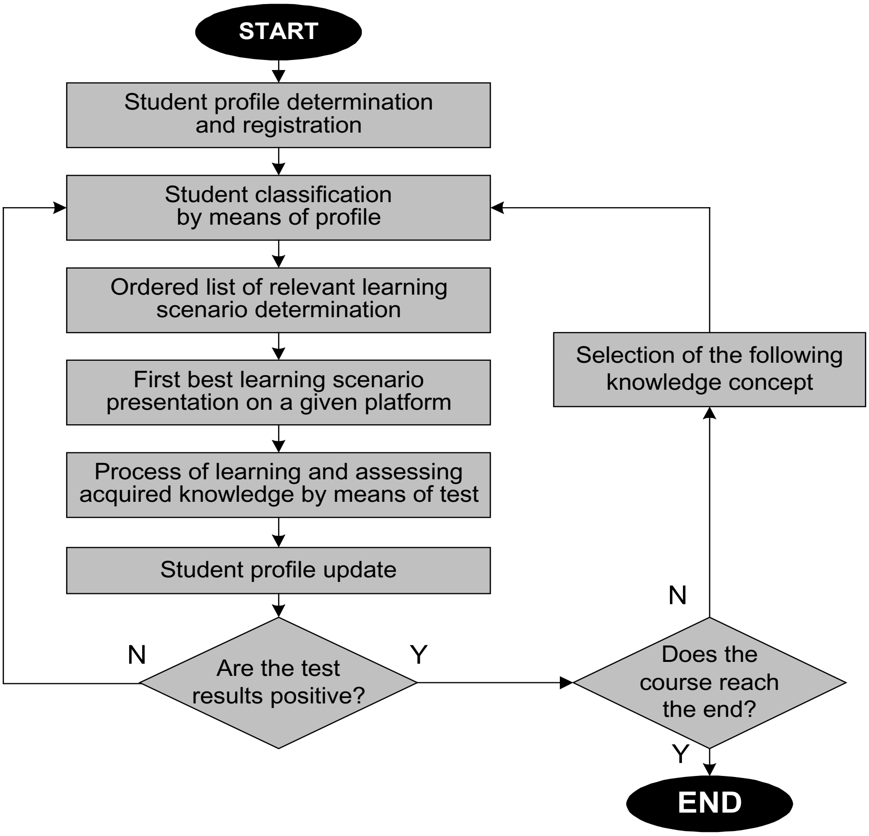

START
Student profile determination and registration
Student classification by means of profile
Ordered list of relevant learning scenario determination
First best learning scenario presentation on a given platform
Process of learning and assessing acquired knowledge by means of test
Student profile update
N
Are the test results positive?
Y
Selection of the following knowledge concept
N
Does the course reach the end?
Y
END

Fig. 2. Architecture for consensus-based learning scenario adaptation.

The overall architecture for the learning scenario adaptation is presented in fig. 2. The result of the adaptation is the learning scenario recommendation for a given student. The scenario contains the collection of pages with corresponding test.

4 Navigational Link Extraction for Learning Strategy

Nowadays the best solution for hyperlink modelling seems to be the XML Linking Language (XLink) [1,3,4]. There are two main link types provided by XLink standard: simple (similar to HTML anchors) and extended. The second ones are allowed to be stored outside the source document what is very useful for adaptive e-Learning content generation. A single `xlink` contains two general XML child elements: `locators` and `arcs`. Locators store references - bindings (URIs) to linked resources and assign them labels and titles. Titles can be later presented to a user as target resource names (as the content of `<a>` element in HTML). In e-Learning systems locators are used for pointing at both scenario media and particular presentation pages. Locators may indicate any element within an XML external document using XPath expression (XPath is the standard retrieval language for XML). In the example shown below these expressions have only the form of identifier references (e.g. `#e0015`, `#p121`) that means pointers to elements containing any attribute of ID type with certain value (`e0015` and `p121` respectively):

```
<link_database xmlns:xlink="http://www.w3.org/1999/xlink"
               xlink:type="extended">
 <name xlink:type="title">Link for scenario</name>
             <!-- resource mapping (binding and indicating) -->
 <media xlink:type="locator"          xlink:title="Video no. 1"
        xlink:href="media_el.xml#e0021" xlink:label="IntroVideo" />
 <media xlink:type="locator"          xlink:title="Video 2"
        xlink:href="media_el.xml#e0015" xlink:label="IntroVideo" />
 <media xlink:type="locator"          xlink:title="Video no. 5"
        xlink:href="media_el.xml#e0024" xlink:label="BasicCourseVideo"/>
             ...
 <page  xlink:type="locator"          xlink:title="Page no. 1"
        xlink:href="pages.xml#p121"     xlink:label="Page1"/>
 <page  xlink:type="locator"          xlink:title="Page no. 2"
        xlink:href="pages.xml#p142"     xlink:label="Page2"/>
             ...
          <!-- arcs: resource connections -->
 <video xlink:type="arc"             xlink:title='Video Basic'
        xlink:from="IntroVideo"       xlink:to="BasicCourseVideo" />
 <video xlink:type="arc"             xlink:title='Video Intro'
        xlink:from="BasicCourseVideo"  xlink:to="IntroVideo" />
 <pages xlink:type="arc"             xlink:title=″Pages′ links″
        xlink:from="Page1"            xlink:to="Page2"
             ...
</link_database>
```

Arcs create connection between all locators with specified label. An arc begins in any resource with the label equal to arc’s `from` attribute value and ends in a resource pointed by `to` attribute. Please note that a single arc can bind many resources simultaneously, because many resources are allowed to possess the same label. For example, the first arc (`Video Basic`) in the link database presented above makes at least two connections: from `Video no. 1` to `Video no. 5` and from `Video no. 2` to `Video no. 5`. The third arc (`Pages′ links`) links from `Page 1` to `Page 2`.

Hyperlinks starting on a source page presented to a user may be merged dynamically with the page content using of the external link database similar to described above and appropriate XSL transformation. We only need to retrieve suitable resources pointed by arcs (fig.3). For particular source page its identifier and identifiers of media resources (content elements) presented on it are known after adaptive scenario selection. Having these identifiers, labels of suitable locators can be retrieved from the link database. All arcs with attribute `from` equal to obtained locator labels point at target resource owing to their `to` attribute. Links of required form (suitable for user platform) can be generated from addresses of these target resources. The link generation, resources and arcs retrieval are done within proper *templates* in an XSLT stylesheet. These templates are then incorporated into the final XSLT so obtained links are outbound [1].

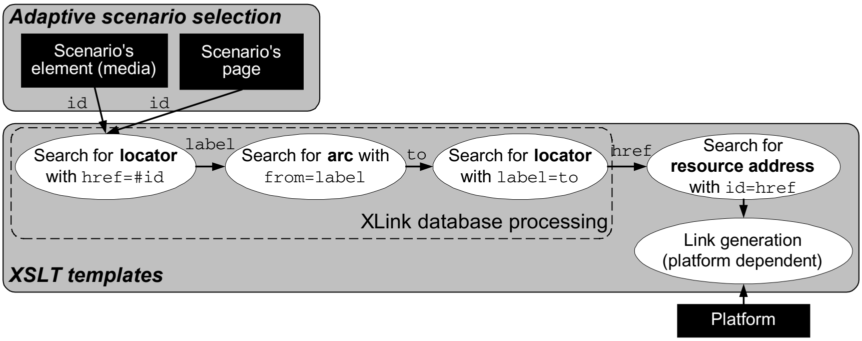

Adaptive scenario selection
Scenario's element (media)
Scenario's page
id
id
Search for locator with href=#id
label
Search for arc with from=label
to
Search for locator with label=to
href
Search for resource address with id=href
XLink database processing
Link generation (platform dependent)
XSLT templates
Platform

Fig. 3. The concept of link generation form XLink database using XSLT

5 Learning Strategy Content Presentation on Different Platforms

A content presented to a student *st* depends on the three issues: layout (*L*), presentation platform (*F*) and learning scenario determined adaptively (s_{st}) in form of the sequence of pages with information content and tests.

The layout reflects esthetic preferences specified directly by a learner that are stored in the e-Learning system. It can be realized during the registration process: the system presents snapshots of all possible layouts to a learner, who selects the one that is looking the best for him or her. These preferences may also be changed before the presentation starts. The set of layouts is fixed and dependent on the e-Learning environment. Thus, layout selection is based on manually selected preferences. However it may also be done automatically by analyzing previous user's e-Learning sessions - the idea similar to link recommendation presented in [6].

The second factor is a platform recognized by the system. This limits presentation possibilities of the particular scenario media types - not all are able to be shown on a given platform. These platforms are defined not only by the hardware (PDA, mobile phone, computer-host, printer) but also by the operating system (Windows, Linux) and even the presentation tool, e.g. browser (IE 6.0, Netscape 7.1). The platform has the biggest influence onto the output format of presentation: HTML (for specific browsers), HTML Basic and WML (for mobile phones), PDF (for printers).

Both the platform and layout determine main transformation process (fig. 4). The system generates (or retrieves from static repository) the independent stylesheet (XSLT) for each pair layout-platform (L_i, F_j). This stylesheet, including so called *templates*, describes the way of data transformation from the source format into the format, which is appropriate for the specific platform. The layout L is responsible only for the location of particular scenario elements on the output page. The transformation language (XSLT) enables to obtain almost any textual format for target documents, including HTML, WML, TXT, XSL FO (XSL Formatting Objects), etc. [5]. From XSL FO documents it is also possible to generate PDF or Postscript files, which are suitable for printers, using free software, e.g. FOP, quite easily.

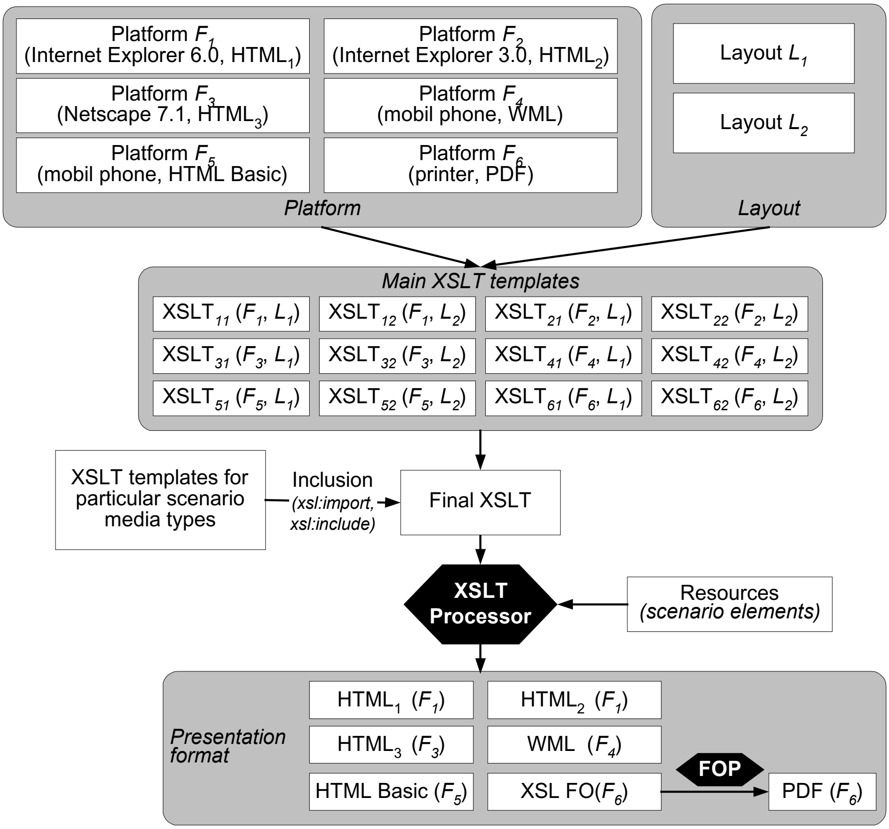


Fig. 4. XSL Transformation for platform and layout

The pair (L_i, F_j) corresponds only to the main stylesheet templates but the page itself includes data specific for particular scenario elements. For each element (media) type exist in the system template set (XSLT) responsible for transformation of this type into output format (fig. 5). Please note that element templates (like link generating templates) are also platform dependent. All these necessary templates are attached to the main stylesheet using XSLT import mechanisms (`xsl:include` or `xsl:import`). Main XSLT stylesheet extended with element and link templates is applied to the resources (especially XML documents) related to given scenario page. The page in platform dependent format is obtained as the output of transformation process.

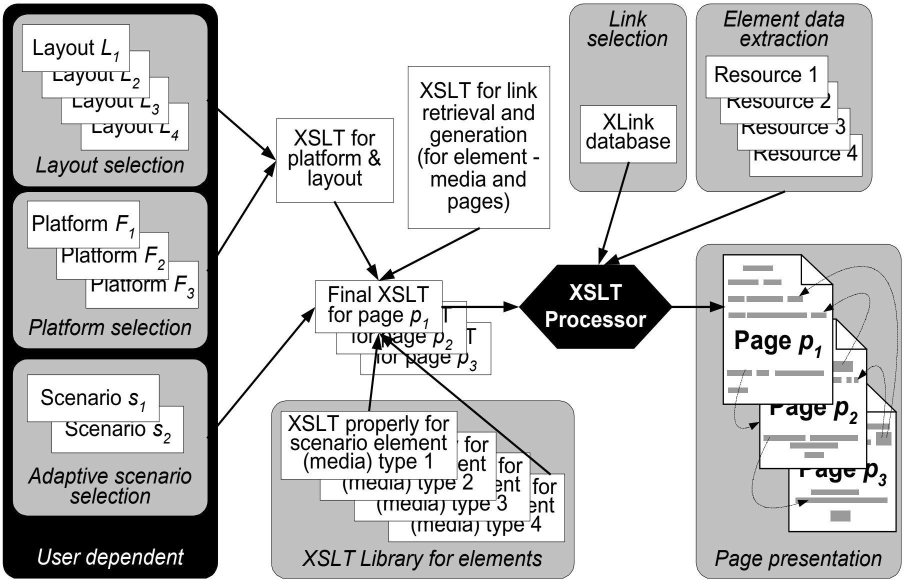


Fig. 5. Content presentation process

6 Conclusion

In the paper the XML-based implementation of adaptive learning scenario is presented. The adaptation of the learning scenario is based on the consensus methods, which recommends for each student an ordered list of learning scenarios. They are ordered with decreasing relevance to the given learner. As some of the media resources, which are placed on pages that are elements of learning scenarios, could not be displayed on some computer platforms, the given scenario is rejected and the next one is considered.

The presented adaptive e-Learning environment architecture is rather simplified so in the future works we shall consider other more complex approaches. We can for example accept such scenarios which contain elements that could not be displayed on the given platform but such media should be replaced by other acceptable resources. This however may cause the decrease of the efficiency of the given scenario so we shall also consider whether the second best scenario is not better than the current one with modifications. The other approach is to consider the system platform in the student profile and to classify them respectively or consider direct interface adaptation [15].

References

1. DeRose S, Maler E, Orchard D (eds) (2001) XML Linking Language (XLink) Version 1.0. W3C Recommendation, World Wide Web Consortium, 27 June 2001, http://www.w3.org/TR/xlink/.
2. Felder R, Soloman B (1998) Index of Learning styles http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS-a.htm.
3. Gwiazda K, Kazienko P (2001) XLink - the future of document linking. Proc. of the Information Systems Architecture and Technology ISAT 2001. Wroclaw University of Technology: 132-139 http://www.zsi.pwr.wroc.pl/pracownicy/kazienko/pub/ISAT2001/XLink.ps.
4. Kazienko P (2002) Finesse of hyperlinks. Tele.net Forum 9: 24-29. (in Polish).
5. Kazienko P, Gwiazda K (2002) Taking XML seriously. Helion, Gliwice (book in Polish).
6. Kazienko P, Kiewra M (2003) Link Recommendation Method Based on Web Content and Usage Mining. New Trends in Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM´03 Conference, Zakopane, June 2-5, 2003, Advances in Soft Computing, Springer Verlag: 529-534 http://www.zsi.pwr.wroc.pl/pracownicy/kazienko/pub/IIS03/pkmk.pdf.
7. Kobsa A, Koenemann J, Pohl W (2001) Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. Knowledge Eng. Rev. 16(2): 111-155.
8. Kukla E, Daniłowicz C, Nguyen NT (2003) Reconciling inconsistency in selection of optimal strategy in multimedia intelligent tutoring systems. Proc. of HERCMA 2003, Athens 24-27 Sep. (to be published).
9. Kukla E, Nguyen NT, Sobecki J, Danilowicz C, Lenar M (2003) A Model Conception for Learner Profile Construction and Determination of Optimal Scenario in Intelligent Learning System. Lecture Notes in Computer Science 2774: 1216-1222.
10. Kukla E, Nguyen NT, Sobecki J (2001) The consensus based tutoring strategy selection in CAL systems. World Transactions on Engineering and Technology Education 1: 44-49.
11. Lennon JA (1997) Hypermedia Systems and Applications – World Wide Web and Beyond. Berlin: Springer – Verlag.
12. Montaner M, Lopez B, de la Rosa JP (2003) A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review 19: 285-330.
13. Nguyen, NT (2002) Consensus systems for conflict solving in distributed systems. Journal of Information Sciences 147: 91-122.
14. Sobecki J, Morel K, Bednarczuk T (2003) Web-based Intelligent Tutoring System with Strategy Selection Using Consensus Methods. New Trends in Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM´03 Conference, Zakopane, June 2-5, 2003, Advances in Soft Computing, Springer Verlag: 105-109.
15. Sobecki J (2003) XML-based Interface Model for Adaptive Web-based Systems User Interfaces. Proceedings of International Conference on Computing Science 2003. Lecture Notes in Computer Science (Springer-Verlag) 2660: 592-598.

Building Ontologies for Interoperability among Learning Objects and Learners

Yevgen Biletskiy[1], Olga Vorochek[2], and Alexander Medovoy[2]

[1] Department of Electrical and Computer Engineering, University of New Brunswick
P.O. Box 4400, 15 Dineen Dr., D41 Head Hall, Fredericton, NB, E3B 5A3, Canada
biletskiy@unb.ca
[2] Department of Software, Kharkiv State University of Radio-Electronics
14 Lenin av., Kharkiv, 61166, Ukraine
relf@kture.kharkov.ua

Abstract. The proposed research is devoted to application of context mediation, an approach for achieving interoperability among semantically heterogeneous databases, for e-Learning. Because the fact that many learning objects are created in various parts of the world and across many cultures these learning objects are semantically heterogeneous, and the delivery of these objects to learners is not effective. The context mediation described in this paper assumes comparison of contexts associated with any learning object and learner, and elimination of semantic conflicts that are detected through this comparison. The comparison of contexts of learning objects and learners uses ontologies as specifications of concepts, their properties and relationships between them in the knowledge domains of courses to be studied. This work is concentrated on the building, contextualization and integration of ontologies, and their use in the context mediation for delivery learning objects to learner's context.

1 Introduction

The presented research is related to application of the Context Interchange approach for e-Learning. Context Interchange is an approach for achieving semantic interoperability among sources and consumers of large-scale, semantically heterogeneous databases [1,2,3]. The main component of this technology is the *Context Mediator*, an intelligent agent that performs data conversions between the sources and consumers. It performs the conversions by comparing contexts *(Import and Export contexts)* of any two databases (source and consumer) involved in data exchange and eliminating semantic conflicts that are detected [1,2]. The Context Interchange approach is currently being pursued by a research group at MIT [3].

The comparison of contexts, detection and elimination of semantic conflicts in the Context Interchange technology is performed by the Context Mediator using *Common Ontology,* which specify concepts and relationships between them, which are common (or understood) for any source or consumer involved in data exchange. Gruber [4] defines ontology as "an explicit specification of a conceptualization". Generally, the research presented in this paper is related to development of the delivery technology

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 977-986, 2004.

for transporting learning objects with various contexts to learners using the Context Interchange approach.

Within the Context Interchange (or context mediation) approach (fig.1) the *Learning Objects*, such as course outlines, course notes, textbooks, pictures, and visual presentations, are the semantically heterogeneous data sources loaded from the local digital library or Web. The Learning Object *MetaData* contains the semantic description of the Learning Object that constitutes *Export Context* of the Learning Object. The *Learner* is the data consumer. The *Learner Context* is the description of mental and cultural characteristics and standards of the Learner. The *Learner Context* constitutes *Import Context* of the Learner. The MetaData and Learner Context can be described by the same language or formal system and usually contain characteristics, such as language, measuring system, conventional data types, cultural features, etc. The Context Mediator compares the Export and Import contexts (the MetaData and Learner Context) and eliminates semantic conflicts that are detected through the comparison process. As a result of the elimination of these conflicts, the Context Mediator generates a sequence of context conversions that are necessary to represent the Learning Object in a context (or a view), which is understandable for the Learner. This Learning Object is delivered to the Learner.

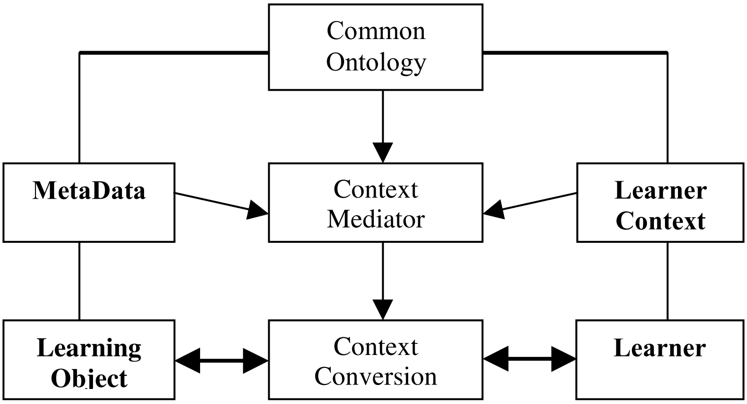


Fig. 1. Context mediation for delivery of *Learning Objects* to *Learner*

This comparison of the contexts is based on the Common Ontology, as specification of the Context Mediator's knowledge about Learning Objects and knowledge domains, which these Learning Objects belong to. These descriptions should be conventional (or standard) for any Learning Object database and independent of contexts of Learning Objects and Learners. The problem with applying the Context Interchange technology to large-scale Learning Object databases is that the Learning Objects can belong to different domains of knowledge. That is why it is necessary to build and integrate the domain ontologies of the Learning Objects involved in delivery to Learners with the goal of building a domain of Common Ontology. This work is devoted to conceptualization learning materials to build their domain ontologies, contextualization and integration of the ontologies, and their use in the context mediation for delivery learning objects to learner's context. The particular applied task presented in this work is conceptualization of course outlines, building ontologies of the course outlines, their literature sources and knowledge domains.

2 Models and Methods of Building Anthologies for Context Mediation

The definitions and basic principles and criteria for the design of ontologies are introduced in [4,5]. Some existing methodologies for building ontologies are introduced and reviewed in [5,6]. The *Skeletal Methodology* by Uschold and King [5,6,7], which provides general guidelines for developing ontologies, is assumed as basic for the task of building Common Ontology in the context mediation approach for achieving interoperability among learning objects and learners. This methodology consists of four stages: identification of purpose, building the ontology, evaluation and documentation. The stage 2 of this methodology is reasonable for building the Common Ontology because it fits in the previously used methodology for building contexts of databases [8]. The process of building the Common Ontology assumes three steps [7]: ontology capture, coding ontology using a formal language, and integrating existing ontologies. This work is concentrated on the process of building ontologies of learning objects following the Skeletal Methodology and implementing them in the context mediation.

The basic knowledge representation model for formalizing ontology in this work is *ontological graph* (*ontograph*) [8]. The ontological graph is *multilevel*, because during the acquisition and forming ontological knowledge the installation of steady associations is happening. A *steady association* [8] is a set of ontological concepts and relationships among them unambiguously describing some knowledge. The steady association is considered as a new concept on a higher level of the ontograph. Because the ontograph is multilevel it is necessary to link its levels. A *transition matrix* described in [8] is used for formalizing the transitions between levels.

The task of searching semantics of concepts consists of the searching chains of semantic links between the corresponding concepts of integrated ontologies. In terms of the context mediation, domain ontology – is a set of object and relationships between them, which are mapped into conceptual model of a data source. Context describes relationships between particular data and its semantics (concepts) in the particular databases. The following context and ontological concepts are used for the context mediation approach: *basic ontology* (BO) – is a basic ontological graph formed for any source or consumer; *integrated ontology* (IO) – is a result of integration of basic ontologies; *steady associated ontology* (AO) – is a resulting ontology with steady associations - this is the higher-level ontology; *contextualized ontology* (CO) – is an ontology with an attached context of any source of consumer involved in data exchange; *key context* (KC) – a set of key concepts and relationships between them for a domain of sources and consumers involved in data exchange; *conceptual context* (CC) - is defined for a fragment of a data source. Both contexts and ontologies are organized as hierarchy. Based on the hierarchy of ontologies (fig.2), it is possible to describe a process of forming an integrated ontology:

$$IO = CO + KC = CO1 + CO2 + \Sigma CC = AO1{*}CC1 + AO2{*}CC2 + \Sigma CC \qquad (1)$$

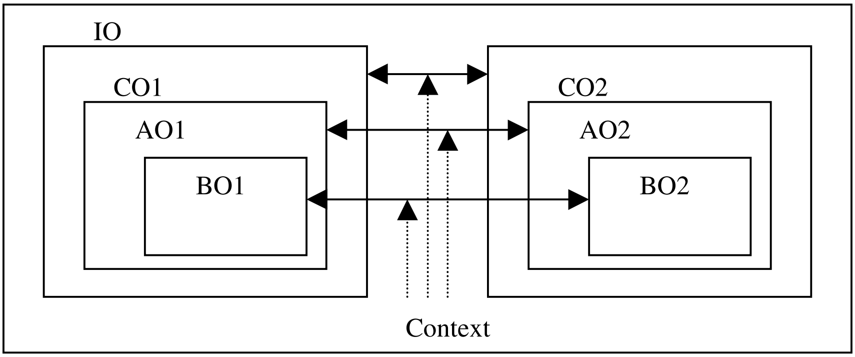


Fig. 2. Ontological hierarchy

Thus, for finding an integrated ontology it is necessarily: to form basic ontologies of the systems involved in data exchange, to resolve ontological conflicts between contextualized ontologies and to find steady associations. The process of forming basic ontologies and principles of integrating ontologies are described in [6]. The contextualization of ontology consists of finding hidden relationships between ontological objects, analyzing clear relationships and linking them. Within this process the key context (KC), which defines key concepts of a domain, can be a criteria for possibility of integrating potentially incompatible systems. If key contexts of two systems involved in data exchange are compatible ($KC1 \equiv KC2$ *or* $KC1 \in KC2$), then finding interoperability uses classic search of ontological links between objects. If key contexts are partially or fully incompatible ($KC1 \approx KC2$ *or* $KC1 \neq KC2$), then indirect methods of finding interoperability are used [8].

3 Context Mediation for Forming Learning Objects to Be Delivered to Learner's Context

In terms of context mediation the process of forming learning objects for learners consists of four stages (fig. 3):

1. Pre-processing – forming a set of object participating in the integration process, definition of data sources (learning objects) and consumers (learners) depending of goals of the integration process.
2. Conceptualization – forming a basic ontological graph of interoperability without consideration of heterogeneous structure of data sources.
3. Contextualization – determining contexts of interoperability and eliminating data conflicts caused by structural and semantic heterogeneity.
4. Finding interoperability among the integrated objects.

The scheme of context mediation adapted to the task of forming a set of learning objects for a particular topic is represented in figure 3. In frame of this task it is necessarily to find a map of a set of files to the set of topics considering literature sources. Thus we process the classes of objects (topics of a course outline, literature, and a set of files of existing learning objects) for finding interoperability among them.

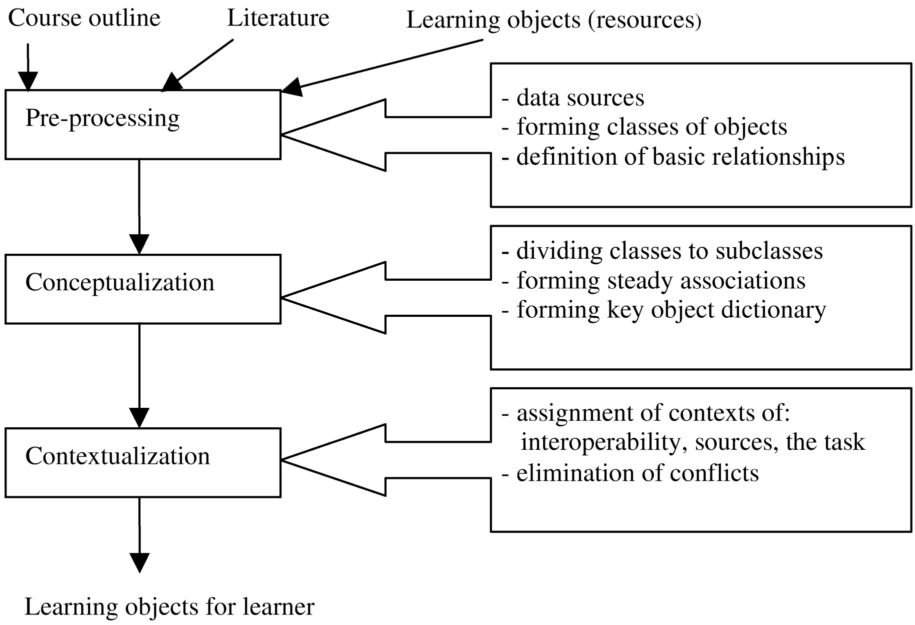

Course outline
Literature
Learning objects (resources)
Pre-processing
- data sources
- forming classes of objects
- definition of basic relationships
Conceptualization
- dividing classes to subclasses
- forming steady associations
- forming key object dictionary
Contextualization
- assignment of contexts of: interoperability, sources, the task
- elimination of conflicts
Learning objects for learner

Fig.3. Scheme of context mediation for forming a set of learning objects for delivery to learner's context

4 Conceptualization of Learning Objects

Let us consider forming ontological graph for the classes of learning objects mentioned above. Assume we have a course outline (that includes the course description, a list of topics and a list of literature) and a set of learning objects for the course (possibly incomplete, complete or redundant). These materials can be structurally and/or semantically heterogeneous. That is why it is necessarily to build context-ontological description of the learning materials.

In terms of the multilevel ontological graph there are two types of ontological objects: description of the basic level O_{01} (i.e. a course outline) and set of learning object files $O_{02}=\{M\}$. The intersection of these objects provides the learning objects to the learning process. Due to heterogeneity of learning objects this intersection could be incomplete, that is why the ontological objects require contextualization. The context C_{01} for the O_{01} defines who proposed the outline, and when it was done. The context C_{02} for the O_{02} is defined by information sources, which participate in forming the set of learning object files.

Assume there are *n* specified documents $\{M\}=\Sigma Mi,i=(1,\ldots,n)$. The basic level of conceptualization allows building the scheme of basic interoperability among learning resources and outlines specifying their use (fig.4).

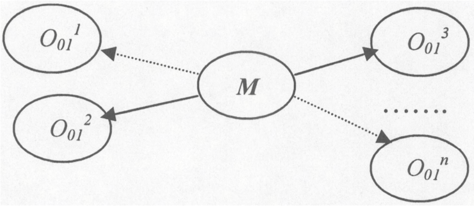


Fig.4. Scheme of basic interoperability. Solid line – the course uses all materials of the learning object, dotted line – the course uses a part of materials of the learning object

The next step of decomposition is building of ontological graphs for each specifying document (course outline). The lists of course topics *{T_i}* and literature *{L_i}* are considered as ontological objects for the O_{01}. Thus, we obtain two sublevels: topics and literature (fig.5). These sublevels allow to review the scheme of use of sources, and then to evaluate completeness of delivery of learning materials for courses. Components of these sublevels are interconnected and describe the sequence of study of the material.

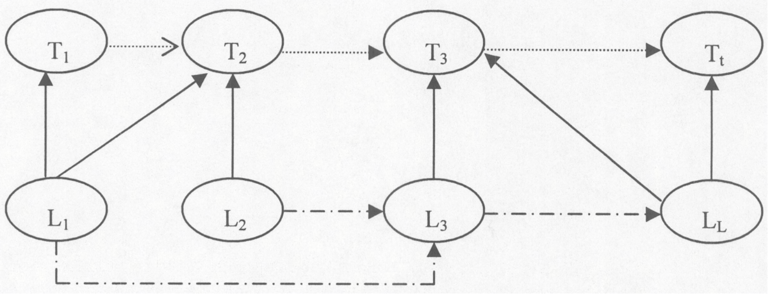


Fig.5. Reallocation object on levels

The next step of decomposition can happen according to two scenarios. The first scenario is analysis of the set of resources without consideration of initial links to the list of literature. In this case decomposition is conducted in the following order:

1. Decomposition of all sets of materials on subsets related to every topic.
2. Analysis of subsets of literature sources, which are necessary for the topic.
3. Finding intersections of subsets *{M}* and *{L}* for discovering required sources of learning objects .

This scenario is preferable if only one course is under consideration. If integration of several courses is necessarily, the second scenario of decomposition is used:

1. Analysis of the intersection of the set of literature sources and the set of learning objects, and forming subsets of primary *{M_1}* and secondary *{M_2}* materials. In this case *{M_1}* is the object of steady associations of appropriate literature sources (the further processing will be applied to the materials excluded from this subset).
2. Further decomposition of *{M_2}* by the same way, etc. (fig.6). This approach allows keeping track accessibility of suggested literature and reducing number of relationships in the graph.

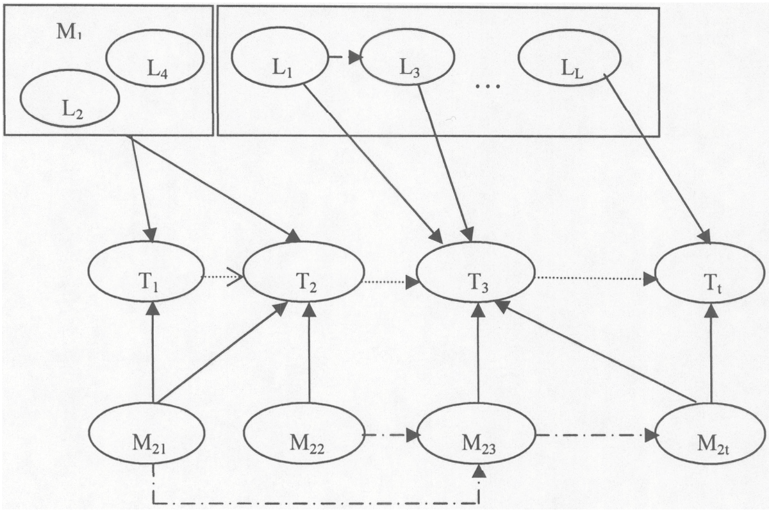


Fig.6. Ontological graph of course description

As a result of conceptualization of learning resources, we have an ontological graph representing conceptual structure of a course including primary scheme of interoperability and approximate scheme of distribution of learning objects among topics of the course. For achieving maximum level of interoperability further contextualization of the graph is happening.

5 Contextualization of Ontologies and Resolving Ontological Conflicts

Contextualization means the process of discovering hidden relationships of interoperability between ontological objects and analyzing clear relationships set up on the conceptualization stage. Objects of the obtained ontological structures can be compatible, but also can conflict. The goal of contextualization is obtaining ontological structure excluding possibility of semantically conflicting description of information. The contextualized ontological graph should accurately describe semantics of ontological domains. The contexts of learning objects and learners determine conditions for interoperability among information sources (course outlines, literature and learning objects) and consumers (learners and course outlines). Course outlines can be both information sources and consumers depending on the particular use.

For discovering types of data conflicts it is necessarily to consider semantics of classes of objects used in the system and types of conflicts between objects of these classes. Semantics of ontological objects describing themes of a course is determined by topics of the themes, subtopics, goals of the course, link to literature and other sources, key terms. Semantics of ontological objects describing literature is determined by some characteristics, such as author, name, year of publishing, etc.

Semantics of learning objects (files) is determined by their internal content and the contexts they were created in. There are the most complex objects for integration due to their structural and semantic heterogeneity. Conflicts of classes are being discovered during the process of building ontological structures of one of the classes. There are structural conflicts caused by conflicting relationships in the ontology to be built (for example, ambiguous description of goals of themes, different sequences of themes, etc.). Conflicts of ontological structures of literature sources are conflicts of generalization caused by incorrect distribution of literature among themes and low level of detailing the literature sources. Conflicts of interoperability are being discovered during the process of integrating ontological structures of course outlines, literature and learning objects.

According to the context hierarchy there are several types of contexts used for contextualization of learning materials. Interoperability context determines conditions for links of interoperability, namely conditions that determine equivalence of semantics of two ontological objects. Identity of semantics of two ontological objects can be set up only if all levels of contexts are semantically identical (including objects of steady associations). Task context is formed for all objects of an ontological graph. The task context determines conditions for interoperability of objects, which belong to the same class within an ontological graph (for example, sequence of topics or key terms for a theme and their intersection). Source context is defined for learning resources (for example, location of a file and its profile, knowledge domain of a resource, author of a resource and authors profile, etc.).

The process of elimination of semantic conflicts is sequential, and the sequence depends on types of conflicts. The biggest number of errors occurs because incorrect processing names of objects (naming conflicts). Conflicts of this type (synonyms and homonyms) are eliminated basing on the conceptual contexts of ontological objects. A conceptual context of ontology is a description of values of ontological objects. For example, for ontological objects of learning materials it could be: *CC={source, author, type}* (where *source* – name of source of an ontological object; *author* – author of the source; *type* – type of the object). If an object is complex: *CC={name, {set_author}}* (where *set_author* – set of authors).

After contextualization of ontology it is necessarily to resolve ontological conflicts to build steady associated ontology. Assume two sets of objects are built: $O1=\{o_{1i}\}$ – describes a set of ontological objects of the first system involved in data exchange, $O2=\{o_{2j}\}$ – the second. Assume that relationships between objects are defined, and potential links of interoperability are found. The first step of resolving ontological conflict is eliminating homonyms. Assume we have two potentially interoperable objects $o_{1i}\in O1$ и $o_{2j}\in O2$. If conceptual contexts $CC1\equiv CC2$, the probability of homonymy is very low, that is why the link between these objects is correct. If $CC1\neq CC2$, homonymy exists.

For example, we have two ontological fragments (fig.7) with potential interoperability, equal conceptual contexts for the objects #3 and different conceptual contexts for the object #1. Thus we have the following set of conceptual contexts:

$CC1_1 = \{name = 1, author1, type1\}$, $CC1_2 = \{name = 2, author2, type2\}$,
$CC1_3 = \{name = 3, author3, type3\}$, $CC2_1 = \{name = 1, \{set\_author1\}\}$,
$CC2_2 = \{name = 4, \{set\_author2\}\}$, $CC2_3 = \{name = 3, author3, type3\}$.

There are two potential links:

Pot1: $O1_1$ -> $O2_1$, $CC1_1 \neq CC2_1$ and *Pot1:* $O1_3$ -> $O2_3$, $CC1_3 = CC2_3$.

In this case objects $O1_1$ and $O2_1$ have the same name but different conceptual contexts, that is why the link Pot1 is excluded and one of the objects is renamed to *1_hom* (this renaming is applied only for ontological object but not for physical data structure). This renaming allows processing homonymic objects independently.

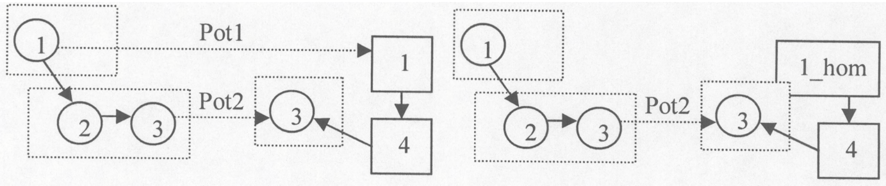


Fig.7. Homonymy and its elimination

Analysis of synonymy is more complicated because synonymy of two objects in different ontologies assumes full coincidence of conceptual contexts and full coincidence of links to other objects. The coincidence of conceptual contexts only assumes potential synonymy; that is why analyzing synonymy is conducted by the following way: (1) extraction of an integrated object of the first ontology; (2) forming a set of links from this object to other objects (mask of interoperability); (3) comparing this mask with every object of the second ontology. Coincidence means synonymy; this is an ideal case because interoperability is achieved, and steady association can be installed (fig.8, first picture, objects #2 and #6). Otherwise we have partial synonymy – it means ontological conflict exists and should be eliminated. In the case of partial synonymy (fig.8, second picture) a key object #5 of the second ontology (part of its link is overlapped by the mask) is split on two objects: with overlapped (#5) and not overlapped (#5_{link}) links. Thus, a quasi-steady association is installed between the integrated objects #2 of the first ontology and the object #5 of the second one. A link of "neighborhood" is installed between objects #5 and #5_{link}.

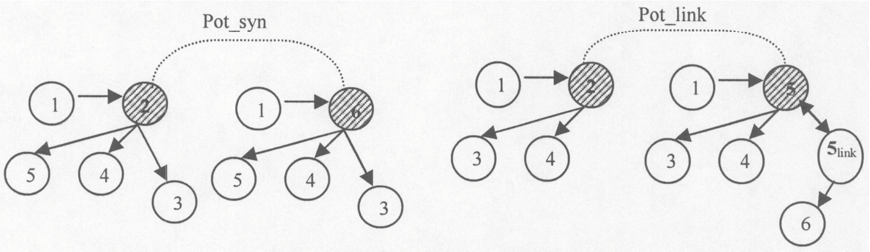


Fig.8. Synonymy and partial synonymy

An example of conceptualization of course outlines and examples of elimination of ontological conflicts through context conversions are described in [9]. The next and the last step of building Common Ontology is the search of meaningful ontological chains between integrated domain ontologies. Classic and genetic algorithms can be used for the search. The methods of integration ontologies are described in [8].

6 Conclusion

This paper describes a model of the process of building the domain of Common Ontology, which is used in the context mediation for achieving interoperability among semantically heterogeneous learning objects and learners in the e-Learning systems. This work is concentrated on the conceptualization learning materials to build their domain ontologies, contextualization and integration of the ontologies and their use in the context mediation for delivery learning objects to learner's context.

References

1. Goh, C. H., Madnick, S. E., and Siegel, M. D. Context interchange: overcoming the challenges of large scale interoperable database systems in a dynamic environment. *In Proceedings of the Third International Conference on Information and Knowledge Management* (Gaithersburg, MD, Nov 29--Dec 1 1994), pp. 337--346.
2. C. Goh, "Representing and reasoning about semantic conflicts in heterogeneous information systems", Ph.D. Thesis, MIT Sloan School of Management, 1996.
3. Firat A., Madnick St., Grosof B. Financial Information Integration in the Presence of Equational Ontological Conflicts, *MIT Sloan Working Paper No. 4408-02; CISL Working Paper No. 2002-16*
4. Gruber T., (1995), Towards Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer studies, 43 (5/6):* 907 -- 928.
5. M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, 11(2):93--136, November 1996.
6. M. Fernandez Lopez, Overview of Methodologies for Building Ontologies. *Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, IJCAI-99.* Stockholm, Sweden. August, 1999
7. M. Uschold and M. King. Towards a methodology for building ontologies. In *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995
8. Ye.V.Biletskiy, Z.V.Dudar, O.G.Vorochek. Integration of Ontologies for the Meta-Context Mediation. *IASTED International Conference on Artificial Intelligence and Soft Computing,* ASC 2003, Banff, Canada
9. Y. Biletskiy, O. Vorochek, O. Biletska, and A. Medovoy. Conceptualization of Learning Objects for Delivery to Learner's Context. *5th International Conference on Information Communication Technologies in Education,* Samos Island, Greece, 2004 (to appear).

Comparison of Different Coordination Strategies for the RoboCupRescue Simulation

S□bastien Paquet, Nicolas Bernier, and Brahim Chaib-draa

DAMAS Laboratory
Computer Science and Software Engineering Department
Laval University, Qu□bec, Canada, G1K-7P4
{spaquet;bernier;chaib}@damas.ift.ulaval.ca

Abstract. A fundamental difficulty faced by cooperative multiagent systems is to find how to efficiently coordinate agents. There are three fundamental processes to solve the coordination problem: mutual adjustment, direct supervision and standardization. In this paper, we present our results, obtained in the RoboCupRescue environment, comparing those coordination approaches to find which one is the best for a complex real-time problem like this one. Our results show that a decentralized approach based on mutual adjustment can be more flexible and give better results than a centralized approach using direct supervision. Also, we have obtained results showing that a standardization rule like the partitioning of the map can be helpful in those kind of environments.

1 Introduction

Cooperative multiagent systems in which agents must interact together to achieve their goals is a very active field of research [13]. A fundamental difficulty faced by such systems is to find how to efficiently coordinate the actions of different agents (independent software entities) to make them help each other, instead of harming each other [3,4,6]. The coordination in multiagent systems is the process used to manage the dependencies between different activities [5].

There are three fundamental processes to solve the coordination problem [7]: mutual adjustment [11,12], direct supervision and standardization [9,10]. Mutual adjustment means that each agent is trying to adapt its behaviour to improve the coordination. Direct supervision, means that there is one agent that can send orders to other agents. Finally, standardization means that there are some social laws enforcing the coordination among the agents.

In this paper, we present some tests comparing a decentralized approach (mutual adjustment) against a centralized approach (direct supervision). We also have made another test comparing the mutual adjustment with the standardization approach. The results presented have been obtained in the RoboCupRescue simulation environment. This is a real-time environment where agents, representing firefighters, policemen and paramedics have to cooperate to save people after a natural disaster in a city [1,2].

In RoboCupRescue, the coordination is very important because there are strong dependencies between agents activities. For instance, a firefighter cannot extinguish a

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 987-996, 2004.

fire if the roads were not cleared by a policeman to enable him to reach the fire. In fact, for a complex problem like the RoboCupRescue, it is not obvious to see which coordination strategy is the most effective. It is why we have made some tests to compare different strategies and to find which one is the best for this kind of application.

This paper compares the usefulness of decentralisation on two different tasks: extinguishing fires and rescuing civilians. In addition, it studies the usefulness of partitioning the environment when the tasks are really dynamic and only one agent has to be assigned to a specific task. The partitioning can be seen as a social law [9] dictating the responsibilities of each agent. Before explaining those strategies, we firstly present our test environment, the RoboCupRescue simulation.

2 The RoboCupRescue Simulation: A Complex Environment

The goal of the RoboCupRescue simulation project is to build a simulator of rescue teams acting in large urban disasters [1,2]. This project takes the form of an annual competition in which participants design rescue agents trying to minimize damages, caused by a big earthquake, such as civilians buried, buildings on fire and blocked roads. In the simulation, participants should have approximately 30 to 40 agents of six different types to manage:

- *FireBrigade*: Their goal is to extinguish fires.
- *PoliceForce*: Their goal is to clear roads to enable agents to circulate.
- *AmbulanceTeam*: Their goal is to transport injured agents and search in shattered buildings for buried civilians.
- *Center agents*: There are three types of center agents: *FireStation*, *PoliceOffice* and *AmbulanceCenter*. The only actions those agents can make are to send and receive messages. A center agent can read more messages than a platoon agent (moving agents in the simulation), thus they can have a better global view of the situation, so center agents can serve as information centers and coordinators for their platoon agents.

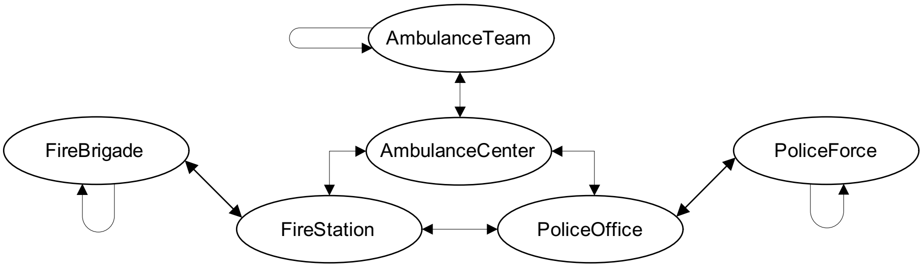


Fig. 1. Communications' organization in RoboCupRescue. Links between types of agents indicate that a communication is possible by radio between those types of agents.

Fig. 1 shows the communications' organization between the different types of agents in the RoboCupRescue simulation. This organization limits the liberty of communication between agents. As we can see, it takes at least three steps for a message to go from a *FireBrigade* agent to a *PoliceForce* agent.

The RoboCupRescue environment is complex because it imposes some hard constraints such as:

- a real-time constraint on the agents' response time because all agents have to be able to reason in less than 500 ms,
- the agents' perceptions are limited,
- the number of messages that an agent can send and hear is also limited.

In the simulation, each individual agent receives visual information of only the region in its surroundings. Thus, no agent has a complete knowledge of the global state of the environment. Therefore the RoboCupRescue domain is in general, collectively partially observable [8]. This means that even if agents are putting all their perceptions together, they will not have a perfect perception of the environment. This uncertainty complicates the problem greatly. Agents will have to explore the environment, as it would not be enough to limit themselves on the visible problems. They will also have to communicate to help each other to have a better knowledge of the situation.

Another difficulty in the RoboCupRescue simulation comes from the fact that the agents are heterogeneous, they cannot do everything by themselves. Therefore, they need to cooperate in order to accomplish their goal efficiently.

Furthermore, agents have to be really careful about the messages they send, because it is really easy to loose messages due to the limitations on the number of messages an agent can hear and also due to the organization presented in Fig. 1.

3 Centralized against Decentralized Decision Making

In this section, we present two different tasks and for each of those two, we present a centralized and a decentralized decision process. First, we define the *FireBrigades*□ task that consists in choosing which fire to extinguish. Then, we present the *AmbulanceTeams*□task that consists in choosing which civilian to rescue.

3.1 FireBrigade Agents

The main task of the *FireBrigade* agents is to extinguish fires. Therefore, the main decision that those agents have to take is to decide which fire to extinguish among all visible fires. To address this, we have developed two different decision processes: one decentralized among all *FireBrigade* agents and one centralized in the *FireStation*.

Decentralized Decision Process. In the decentralized process, each *FireBrigade* chooses the fire it wants to extinguish by itself, so all decisions concerning fire extinguishment is taken in a distributed manner, locally by each *FireBrigade* agent.

Each *FireBrigade* agent is maintaining a list of all the buildings on fire which it knows about. This list is then sorted according to a certain function estimating how good the choice of each building is. With this function, *FireBrigades* are prioritizing certain categories of building. Empirically, we have chosen some weight to define the relative importance of fires□characteristics to make agents prioritize :

- the early fires, with an intensity of 1, because they are generally easy to extinguish and at the border of a zone on fire;
- the fires that put the biggest area in danger, to protect the biggest area;
- the closest fires, because it is faster to reach such fires;
- the smallest fires, because they are easier to extinguish;
- the fires with *FireBrigades* already on it, because it is faster to extinguish a fire in group;
- the buildings that are not concrete reinforced, because those are harder to extinguish.

Centralized Decision Process. In the centralized decision process, practically all decisions are made by the *FireStation* agent. Therefore, the decision process is centralized in one agent. The *FireStation* is informed of the situation, i.e. where the fires are, by messages sent by all agents in the simulation. At each turn, the *FireStation* agent uses a function, similar to the one described in the decentralized approach, to sort all fires according to their importance.

Afterwards, the *FireStation* agent sends a message to each *FireBrigade* containing the two best fire it has identified. Those two fires are seen as orders by the *FireBrigades*, so they obey and try to extinguish the first fire on the list. When the first one is extinguished, they try to extinguish the second one.

The *FireBrigade* agents blindly obey the *FireStation*. The only time they choose by themselves which fire to extinguish is when they do not receive a message from the *FireStation*.

Results. To test those two different coordination approaches, we have made some test runs on three different maps. To study the effectiveness of our agents at accomplishing the extinguishing task, we have recorded the surface burned in each situation. In Fig. 2, we can see two graphics representing the average surface burned during 20 runs on two different cities. In Fig. 2 a), the decentralized approach gives better results than the centralized approach, but on Fig. 2 b), it is the opposite.

Those results show that there is not one approach that is better than the other in all situations. To explain that, we should note that one of the big difference between the two simulated cities is that at the beginning, the roads were more blocked in city a). This has an impact because the center in the centralized approach does not consider the agents□positions, so if the roads are blocked, there is a good chance that the agents have some difficulties to reach the building on fire. Since each agent in the decentralized approach is considering its distance from the fire, it has more chance to reach the chosen fire as it is normally closer.

With the centralized approach, if the roads are not too blocked, the results are better because the center has the best global view and consequently it can choose the best fire to extinguish and the *FireBrigades* are able to reach the building more easily. When the roads are more blocked, the decentralized approach becomes better because each agent chooses closer fires, so it has more chance to reach them.

That shows that the best approach would be to combine those two extremes by having an approach where the decision is made in a centralized way in some situations and in a decentralized way in other situations. The combination and the identification of those situations will be the subject of future research.

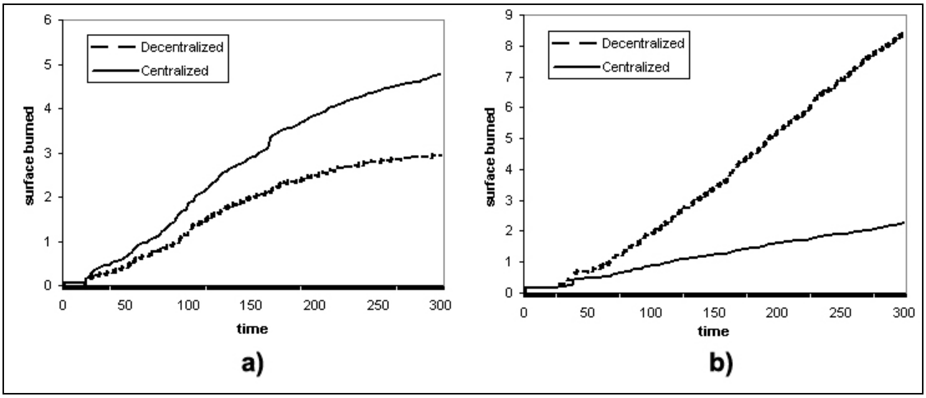


Fig. 2. Average surface burned during 20 runs of two different situations.

3.2 AmbulanceTeam Agents

The main task of the *AmbulanceTeam* agents is to rescue civilians. Therefore, the main decision those agents have to take is to decide which civilian to rescue between all known buried civilians. As for the *FireBrigades*, we have developed two different decision processes: one decentralized among all *AmbulanceTeam* agents and one centralized in the *AmbulanceCenter*.

Decentralized Decision Process. In the decentralized process, all *AmbulanceTeam* agents choose which civilian to rescue by themselves. They use a greedy planning algorithm to try to maximize the number of civilians that could be saved. Each task, corresponding at saving a civilian, has a time length giving the necessary time to save the civilian. This time is the sum of: (a) the travel time to go to the civilian location, (b) the rescuing time and (c) the time to transport the civilian to the refuge. Each task also has a deadline representing the expected death time of the civilian.

The chosen civilian is the one that maximizes the number of civilians that could be rescued after him. To do that, we count for each civilian how many other civilians we would have time to save before its expected death time. The □best one□is chosen by the *AmbulanceTeam* to be rescued. After this civilian has been rescued, the *AmbulanceTeam* chooses the next one with the same algorithm.

Centralized Decision Process. In the centralized decision process, all decisions are made by the *AmbulanceCenter*. At each turn of the simulation, this center agent sends the ordered list of civilians to rescue to the *AmbulanceTeams*. The center agent determines which civilians to save and in which order by using the same algorithm described before for the decentralized process. The only difference is that the center estimates the traveling time to be two times the time to reach a refuge from the civilian location. It has to do an estimation because it doesn□t know the position of each *AmbulanceTeam*, so it does not know the distance from the *AmbulanceTeam* to the civilian. At each turn, the center agent determines the two best civilians to rescue and sends this information to each *AmbulanceTeam*. Since all *AmbulanceTeams* receive the same messages, they are always rescuing the same civilian. This is a good behavior because they work faster when they work together. For example, if one *AmbulanceTeam* agent is rescuing a civilian, it could take 30 minutes, but if the are two rescuing the same civilian, it could take 20 minutes and with three it could take 15 minutes. When they are in group, they can dig faster to find the civilian trapped in the building.

Results. To test those two approaches, we have executed the simulation 20 times on three different maps and we have recorded the total *HP* of all agents in the simulation. The *HP* represents the healthiness of an agent. If the *HP* is 0, the agent is dead.

Fig. 3 shows the average of the total *HP* for the centralized and the decentralized approaches. As we can see, the two approaches are quite similar. In our two other maps, the results are even more identical. One reason for this could be that the *AmbulanceTeam* agents have practically the same vision of the situation as the *AmbulanceCenter*. This is because they receive almost all the messages concerning the civilians. Therefore, since they are choosing the civilians based on the same function they have a good chance to choose the same civilian, thus generating the same behavior as for the centralized process.

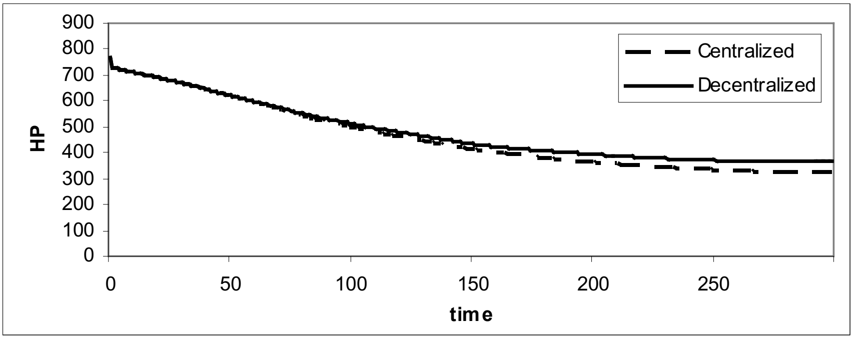


Fig. 3. The average total *HP* for the centralized and decentralized approaches.

The little difference in favor of the decentralized process could be explained by the better individual choice of civilians at the beginning. In the decentralized way, each *AmbulanceTeam* agent chooses a closer civilian, so it can have more chance to be able to reach it if the roads are blocked. This distinction is rapidly lost because after a short time, the *AmbulanceTeam* agents can move more freely and since they all have approximately the same view of the situation, they almost choose the same civilian every time, thus generating the same behavior as in the centralized process.

4 Partitioning the Environment

In this section, we present our results showing the impacts of partitioning the environment. In fact, this simplifies the coordination, because each agent has a region for which it is responsible for. To illustrate that, we present some results obtained with our *PoliceForce* agents in two different settings: with sectors or without sectors.

PoliceForces are playing a key role in the rescue operation by clearing the roads, thus enabling all agents to circulate. Without them, some actions would be impossible because all other agents would be indefinitely blocked by roads' blockades. Therefore, it is really important for them to be fast and efficient. For those agents, we have developed two strategies: one where each agent is free to execute all tasks, no matter where they come from, and an other strategy where some agents are responsible for a specific sector assigned to them at the beginning of the simulation.

4.1 Partitioning in Sectors

For this strategy, we have divided the map in nine sectors to help agents with the task allocation problem. So, at the beginning of the simulation, when the agent receives the information of the city map, it begins by dividing the map in nine homogeneous sectors and sends its position to the *PoliceOffice*. When the *PoliceOffice* has received all the positions of the *PoliceForce* agents, it assigns a sector to the nine agents that are closer to the center of a sector. Thus, there is one and only one *PoliceForce* affected to a sector and this agent has the responsibility of this sector. By doing so, we have divided our *PoliceForces* in two groups: those with a sector and those without a sector. Therefore, even in the strategy with sectors, we have some agents that do not have a sector because there are more agents than sectors. Those agents without a sector are in charge of clearing the roads around the closest fire. Here is a list of the strategies, in their priority order, that *PoliceForce* agents follow:

1. Unblock other agents in its sector.
2. Clear roads between fires and refuges in its sector.
3. Clear roads around refuges in its sector.
4. Clear all the roads in its sector.

Thus, the highest priority task of a *PoliceForce* agent with a sector is to help the other agents in its sector. This is very important, because *FireBrigades* and *AmbulanceTeams* would not be able to do their tasks if they are blocked. Therefore, when an agent is blocked it sends a message to the *PoliceForces* indicating its position. When it receives the message, the *PoliceForce* responsible for this sector adds it to the list of roads to clear. When a *PoliceForce* agent has to make a decision, it chooses the closest road to clear on this list.

When the preceding list is empty, the *PoliceForce* agents with a sector try to open the roads for the *FireBrigade* agents by clearing all roads between a fire in the sector and the closest refuge. Notice that the refuge can be in its sector or not. This is a proactive action to help the *FireBrigades*, because there is a good chance that those agents would ask for a road clearing. *FireBrigades* have a good chance to use those roads because they have to refill their tank at the refuge, so they are often going from a fire to a refuge and in the other direction too. This strategy helps to reduce the communications and the agent movements.

Afterwards, *PoliceForce* agents are clearing roads around refuges. They clear all roads in a perimeter of 40 meters around the refuge, if it is in their sector. This is also a proactive action, because refuges are intensively used by the *FireBrigade* and the *AmbulanceTeam* agents.

When the three preceding tasks have been done, *PoliceForces* clear all the roads in their sector. They first calculate the best path to visit all the roads in their sector. After this, they will follow this path and clear all blocked roads on their path. Thus, when all those tasks are done, all roads on the map have been cleared.

4.2 Without Partitioning

In this case, *PoliceForce* agents have only two tasks. The first one is to clear roads to help the other agents. When they receive a message asking for a clearing, they add it to the list of roads to clear. From this list, they choose the closest road to clear. Since they do not have any sector, they try to help all agents, no matter where they are on the map.

The second task is to clear roads on the path from a fire to a refuge. Unlike the other type of *PoliceForces*, they are not restrained to a specific sector, so they choose the closest fire. By clearing the path from a fire to a refuge, they clear the roads around fires and around refuges. This is interesting since they are really important spots to clear to help the other agents to move freely in the city.

4.3 Results

To test those two approaches, we have executed the simulation 20 times on three different maps. We have recorded the width of each road that is blocked in the simulation and the number of time that an agent is asking to clear a road. The graphic of Fig. 4 a) indicates that the *PoliceForce* agents with sectors gave better global results. In this graphic, the score is the total score given by the RoboCupRescue simulation. There is not a big difference, but in average, the partitioning of the map in sectors helped those agents.

On the Fig. 4 b), we can see that the *PoliceForces* with sectors are more efficient at clearing roads, because there are less roads blocked in average at the end of the simulation. The Fig. 4 c) shows that the number of clearing messages with or without the partitioning strategy is the same. One explication might be that *PoliceForce* agents with sectors are clearing all the roads in their sector, but they may clear roads that are not used by other agents. That could be why the number of clearing messages is the same even if there are less roads blocked when the agents are using sectors.

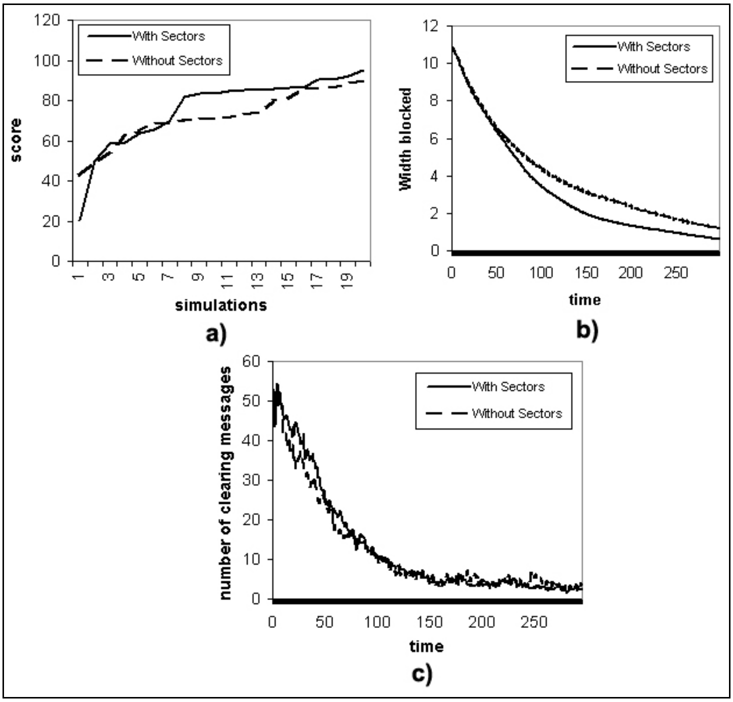


Fig. 4. Comparison of the results obtain for the *PoliceForce* agents with and without sectors. a) the total score for the 20 simulations. b) The average roads□width blocked during the 20 simulations. c) The average number of clearing messages during the 20 simulations.

5 Conclusion

In this paper, we have compared a centralized approach with a decentralized approach on two different tasks. Our preliminary results show that the decentralized approach is better when the situation is more chaotic because the center agents do not know the position of each of their agent. Since platoon agents have some problems reaching the goal of the center agent, they obtain better results when they are deciding by themselves, because they can take their distance from the objective more into account.

In the case of a centralized approach, we obtain better results when the agents are free to move, because in those situations, the distance from the goal is not important. Thus, the better global vision of the center can help to choose the best goal.

For the task of rescuing civilians, we have seen that the difference is not significant between the centralized and the decentralized approach. This happens because the information concerning civilians is less dynamic and the messages are received by all agents. So, every *AmbulanceTeam* agent has sensibly the same view as the *FireStation* agent. Therefore, *AmbulanceTeam* agents often make the same decisions as the center agent.

We have also tested in this paper the usefulness of partitioning the environment in sectors. As we have demonstrated with our results, the partitioning of the map is helpful for a task, like clearing the roads, that is very dynamic. In the RoboCupRescue environment, *PoliceForce* agents can receive a lot of messages asking to clear a road and the partitioning help in assigning a specific clearing task to an agent. Each agent knows the tasks that it is responsible for without any communication. So, there is always one and only one agent executing a specific task. The coordination is assured with those sectors and this can be seen as a coordination by social laws.

In short, we have shown in this paper that when a task is complicated, it is useful to use a decentralized decision making process and also if possible to partition the environment in sectors and assigning an agent to be responsible for each sector.

References

1. Kitano, H., Tadokor, S., Noda, H., Matsubara, I., Takhasi, T., Shinjou, A., and Shimada, S.: Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics (SMC-99)* (1999)
2. Kitano, H.: Robocup rescue: A grand challenge for multi-agent systems. In *Proceedings of ICMAS-2000* (2000)
3. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P., Zhang, X.Q: Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. In *Autonomous Agents and Multi-Agent Systems (To appear),* Kluwer Academic Publishers (2003)
4. Lizotte, S., and Chaib-draa, B.: Coordination in CE Systems: An Approach Based on the Management of Dependencies Between Agents, *CERA: Concurrent Engineering: Research and Applications*, Vol. 5, number 4 (1997) 367-377
5. Malone, T. W., and Crowston, K.: The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, Vol. 26, number 1, March (1994)
6. Martial, F. V.: Interactions among autonomous planning agents. In Demazeau, Y. and Muller, J.-P., editors, *Decentralized AI*. North Holland (1990) 105–119
7. Mintzberg, H.: *The Structuring of Organizations*. Englewoods Cliffs (1979)
8. Nair, R., Tambe, M., and Marsella, S.: Team Formation for Reformation in Multiagent Domains like RoboCupRescue. In Kaminka, G.; Lima, P.; and Roja, R., eds., *Proceedings of RoboCup-2002 International Symposium*, Lecture Notes in Computer Science. Springer Verlag (2003)
9. Shoham, Y., and Tennenholtz, M.: On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the National Conference on Artificial Intelligence*, San Jose, CA. (1992) 276–281
10. Stone, P., and Veloso, M.: Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. *Artificial Intelligence* (AIJ), Vol. 100, number 2, June (1999)
11. Sugawara, T., and Lesser, V. R.: Learning to Improve Coordinated Actions in Cooperative Distributed Problem-Solving Environments. *Machine Learning*, Vol. 33 (1998) 129-153
12. Tambe, M.: Towards flexible teamwork. *Journal of Artificial Intelligence Research*, Vol. 7 (1997) 83–124
13. Wooldridge, M.: *An Introduction to MultiAgent Systems*. Wiley (2002)

Multiple Reinforcement Learning Agents in a Static Environment

Elhadi Shakshuki and Karim Rahim

Jodrey School of Computer Science
Acadia University, Wolfville, NS, Canada
{elhadi.shakshuki;042840r}@acadiau.ca

Abstract. Reinforcement learning agents are used in many domains; however, multiple, benevolent, and communicating reinforcement learning agents are rarely used. This paper explores using multiple reinforcement learning agents in a simple static environment to research methods allowing such agents to communicate during learning. The problem of developing a model whereby reinforcement learning agents can learn faster through the use of communication is explored. This paper develops two agents that are able to communicate with each other and able to learn about their actions using reinforcement learning techniques. The maze environment is used as a testbed. In this environment, the two agents are required to navigate the maze and trade information about the state of each other. This information includes the expected utilities for all actions from that state. To demonstrate the feasibility of the proposed approach, the agents are implemented. In addition, the efficiency of this model is compared with single agents and two independent agents.

1 Introduction

Reinforcement learning (RL) mimics one method animals use to learn. Animals perform actions that lead to positive reinforcement, and they avoid actions that lead to negative reinforcement. In addition to learning by receiving reinforcement from their own actions, animals learn from the experience of each other. RL is described by Bigus and Bigus [1] as, “the most realistic form of learning.”

RL is often used in autonomous agents, but there are few [12] applications for multiple RL systems. In our work, we focus on developing a multi-agent RL system in a game environment. The main objective is to develop and test an environment that consists of two RL agents, where each agent is able to learn from its own trial and error actions and from the trial and error actions of each other.

While much of the work with RL agents□focuses on a single agent, there exists some work with multiple agents. Ming Tan [12] compared independent and cooperative RL agents, and he identified three methods of communication for RL agents. In another work, Whitehead [14] developed two cooperative learning agents: *Learning with an External Critic* (LEC) and *Learning By Watching* (LBW). Whitehead found that both procedures reduce the search space; in addition he found the LBW agents

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 997-1006, 2004.

could perform well when all agents begin na□ve, whereas the LEC algorithm requires a competent critic or supervisor. The watching agent or critic agent is off-line.

A simple maze navigation environment, developed by Perez-Uribe [5] was used as a testbed for this experiment. The environment was chosen because it would not offer an unfair advantage to cooperating agents. RL is a tightly coupled system; this paper addresses an attempt to introduce knowledge communication, in the form of policy sharing, into such a system. This work investigates two cooperative RL agents in the test environment. The agents are given a communication option in addition to their movement options. The learning rate of the two communicating RL agents will be quantified and compared to two independent RL agents, and to one autonomous RL agent.

2 Related Work

Several multi-agent RL systems have been developed; essentially these systems seek to take advantage of multi-agents to reduce the search space. These systems use various methods of allowing communication between multiple agents. A shared whiteboard method is explored with the MALE system [9] that uses an interaction board. The ILS [10] system uses a central controller approach to integrate heterogeneous learning agents. The previous systems used learning from examples. San and Weiss [8] explain that learning is a method for reducing the load of communication among individual agents, and that communication is viewed as a method for exchanging information that allows agents to continue or refine their learning activities. They find that communication offers numerous possibilities to improve learning, but it is not a panacea for learning. Whitehead [14] discussed two systems: a system with *Learn By Watching* (LBW) agents, and a system with *Learning with and External* Critic (LEC) agents. He proposed that multiple RL agents would decouple the learning rate from the state-space size. He found that both procedures reduce the search space. Whitehead theorized the lower bound by which *n* agents can decrease the learning rate as $\Omega\left(\frac{1}{n}\right)$. San and Weiss [8] studied concurrent, isolated reinforcement learning (CIRL), which is similar to an embarrassingly parallel system. They studied domains where other agents affect the dynamics of the environment, and found that the CIRL paradigm is most effective when the agents□actions are not strongly coupled. Weiss [13] studied agents communicating in order to decide on individual group actions. He adopted studies of algorithms similar to that of bucket brigade algorithm (BBA) [13] used in classified learning. The two algorithms he studied are the Action Estimation (ACE) and the Action Group Estimation (AGE). Optimal performance was not attained with either procedure partially because of each agent□s limited local perception, and partly because of the inability of the agents to consider states globally instead of locally. In addition, the algorithms do not consider the cost of communication. Littman [3] described a system of competitive RL agents, where these agents are learning to play against each other in soccer-like game environment. Although, in his approach the RL agents perform well, they do not consistently beat his hand programmed agent.

Prez-Uribe and Sanchez [6] discussed the possibility of a multi-agent three-player card game, but the brunt of their work focused on a single RL neural network.

Makar et al. [4] described a hierarchal approach to multi-agent RL. In his approach he scaled RL using the MAXQ hierarchical framework and used joint action values. The key principle explored was that the value of a parent task could be factored in to the value of subtasks that are independent of joint action values, and the completion cost which depends on action values. This approach requires an appropriate hierarchical model for success. Recently Tan [12] compared cooperative RL agents to independent RL agents and outlined the following three methods of communication for RL systems: (1) Agents can communicate instantaneous information, such as sensations, actions, or rewards. (2) Agents can communicate sequences of episodes. (3) Agents can communicate learned decision policies. Our proposed system allows agents to communicate with learned decision policies.

3 Reinforcement Learning

RL is online, and it combines elements of supervised learning and dynamic programming. In RL, the agent receives a signal or signals from the environment that serve to locate it in a discrete state, which contains all the necessary information for the agent to select its next action. Such states are said to exhibit the Markov (memory-less) property [11]. The agents in our system are equipped with knowledge of their current state, and they can perceive attributes of immediately adjacent squares.

In our proposed system, agents are equipped with the *SARSA*(λ) learning algorithm [11][5]. *SARSA*(λ) is a *Q-learning* algorithm that learns a decision policy based on the state action value function *Q*. This is an active learner where the current state and the next action are represented by a state-action pair. *SARSA*(λ) is a temporal difference (TD) algorithm that updates the value of the i^{th} state-action pair to bring it in equilibrium with the $(i+1)^{th}$ state-action pair [7]. *SARSA*(λ) uses an eligibility trace that controls which state-action pairs are more eligible for learning [5]. To describe the *SARSA*(λ) algorithm, we need to define the following. Assume that s refers to the current state of the system, a refers to the action just taken, $s^{'}$ refers to the next state of the system, $a^{'}$ refers to the current selected next action, α refers to a discounted learning parameter (learning rate), γ refers to a discount factor, r refers to the reward received for doing action a in state s, λ refers to a real value between 0 and 1 used in reducing eligibility trace values, and ε refers to a value less than one, indicating the amount of exploration. The value of a state-action pair is modified based on the reward received from a specific state-action. In order to appropriately map a reward to not only the most recent state-action pair, but to state-action pairs that preceded the current state-action pair, an eligibility trace is kept. The value of the eligibility trace for a specific state-action pair is set to one immediately after the state-action pair occurs. The value is then reduced by a factor of γ^{λ} after each subsequent state-action pair. The *SARSA*(λ) algorithm can be described as follows [5]:

The temporal difference error (*TDerr*) is calculated first to update the agent's knowledge with the current view of the world, using Equation (1). The *TDerr* is the reward received from executing action *a* from state *s* plus the difference of the value associated between the new state-action pair $Q(s',a')$ and the executed state-action pair $Q(s,a)$.

$$TDerr = r + Q(s',a') - Q(s,a) \tag{1}$$

This error is calculated after each transition from state s to state s'. The eligibility for the current state-action pair is set to 1.

$$e(s,a) = 1$$

The *TDerror* is used to update the *Q* values for each state-action pair. The *TDerror* is multiplied by eligibility trace value for the state-action pair, using Equation (2).

$$Q(s,a) = \ Q(s,a) + \alpha \ TDerr \ e(s,a), \quad for\ each\ s \in S,\ a \in A. \tag{2}$$

The eligibility trace values are reduced by a factor of γ^{λ} for each state-action pair, using Equation (3). The updated trace will be used for all state-action pairs not directly executed in the next action.

$$e(s,a) = \gamma^{\lambda} \ e(s,a), \quad for\ each\ s \in S,\ a \in A. \tag{3}$$

During the agent lifetime it always tries to maximize it's rewards before taking an action, using the following learning procedure:

- Select an action such that $Q(s, a)$ is maximum (i.e., with probability 1- ε).
- Execute the selected action *a* and enter s'.
- Receive the reward value *r*.
- Select the likely next action a' for use in updating $Q(s, a)$.
- Update $Q(s, a)$ using *SARSA*(λ).

The RL agents cooperate with each other in trading values for state-action pairs. One of the main key issues considered in RL is the assignment of reward. For this model a *minimum time to goal function* is utilized. An agent receives a reward of -1 for each action other than the goal action and 0 for the goal action. This approach motivates and encourages the agent to find a short path to the goal.

4 Environment

The environment of the RL agents used is as shown in Figure 1. It is a maze environment with a fixed goal location. In this environment, the RL agent navigates the maze starting from square *i* (i.e., initial state) to square *j* (i.e., the final state or goal). It is assumed that the agent will not encounter any terrible traps. This makes the agent's main objective is to learn how to navigate towards the goal without running into walls. Figure 1 shows the agent□s perceptions. These perceptions are limited to the agent□s adjacent four squares. Each of the four squares is represented by one bit, where 1 is assigned for blocked and 0 is assigned for not blocked. For example, the value 1100_2 shown in Figure 1 indicates that the south and the east are blocked, while the west and the north are not blocked. The agents in this environment are able to select from one of four actions. The agent can move one square to the north, south, east or west. In the multi-agent game, the agent can also select a fifth action *ask*(*s*), which allows an agent to request all state action values for state *s*..

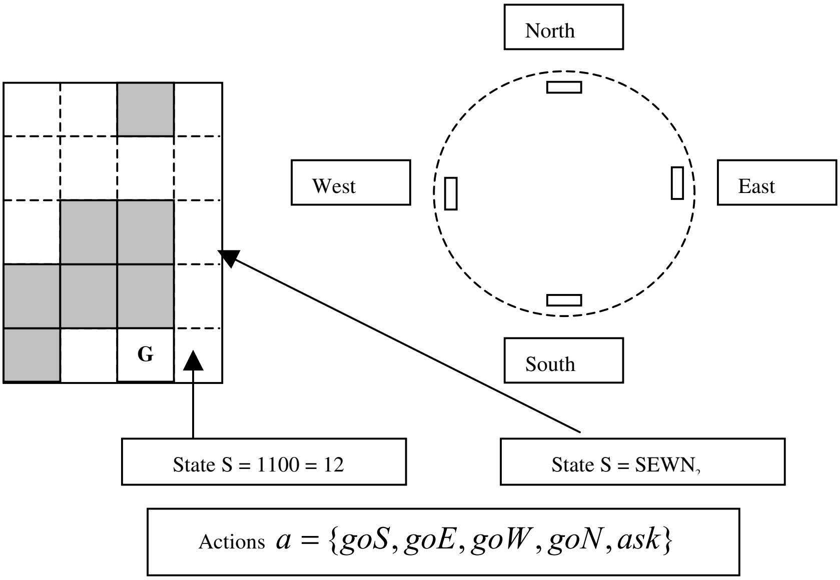


Fig. 1. Game Maze and Agent Perception

This environment is accessible, as it is assumed that the agent can see the adjacent locations. The model the environment object has a static method that updates the agent□s state. The agent has all the required knowledge to select the desirable action. In addition, the environment is: deterministic, the agent can perceive its environment without 100% certainty; static, the maze environment, including the position of the goal square, do not change; and discrete, progression in time is marked by transition from one state to the next.

5 Agent's Architecture

The architecture of the proposed RL agent is designed to navigate a maze environment. Each agent is autonomous, cooperative and able to learn and communicate with other agents to achieve its goals. The agents communicate with each other using a simple version of Knowledge Query Manipulation Language (KQML) [2].

The architecture of system is shown in Figure 2. The system consists of a set of components including: problem solver, sequential control, knowledge update, learning, and communication components. The problem solver observes the state of environment and determines the appropriate next action; in *SARSA*(λ) each state is mapped to a best action by a policy π. The sequential control component synchronizes the discrete time steps, while the knowledge update component update the agents learned policy using the *SARSA*(λ) algorithm. The communication component consists of an inter-agent communication, and environmental perception component.

After a series of trial and error attempts the agent learns a policy π that maps states to effective actions. The agent keeps the learned policy in memory; however, they have no memory of the sequence of past actions.

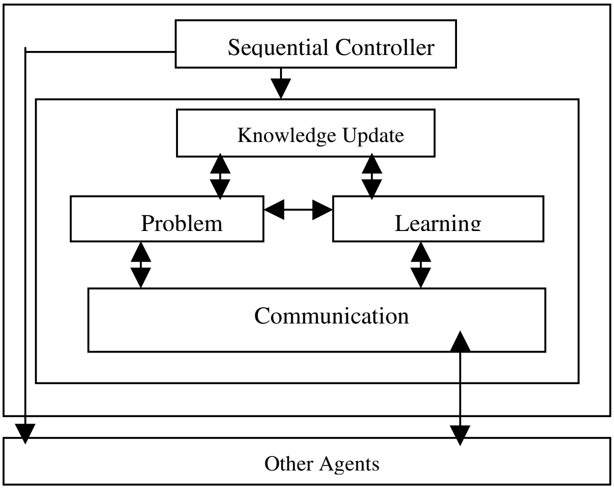


Fig. 2. Agent□s Architecture.

6 Implementation

The proposed system has been implemented using Java; the original code [5] was ported from C to Java, modularized, then multiple agents were introduced. In addition, the agents were integrated with a statistics module. A control module ensures both agents act in parallel. A weighted average function is used to reconcile the knowledge received from communication. The value for each state-action pair received is weighted-averaged with the requesting agent's value for each state-action pair. A weight value *w*, where *w* is a value on the interval [0, 1], is multiplied by the received value, then the value is added to (1 - w) multiplied with the original value. To describe the weighted average function, we define Q"(s, a) as the state-action pair received in response to the *ask()* request. Equation 4 shows the weighted average function. Several values *w* are tested in order to find an optimal value for the given model.

$$Q(s,a) = wQ''(s,a) + (1-w)Q(s,a) \qquad \textit{for each } a \in A. \tag{4}$$

The complete procedure used in by the communicating follows:

- Select an action such that $Q(s, a)$ is maximum (i.e., with probability 1- ε).
- If the action selected is *ask*
    - Request all the $Q(s, a)$ values from the other agent for state *s*
    - Set $Q(s, a)$ to the weighted average of the received and original values
    - Select an action such that $Q(s, a)$ is maximum (i.e., with probability 1-ε).
    - If the action selected is *ask*
    - Select and action *a* at random from the set of actions not including *ask*.
- Executed the selected action *a* and enter s'.
- Receive the reward value *r*.
- Select the likely next action a' for use in updating $Q(s, a)$.
- Update $Q(s, a)$ using *SARSA*(*λ*).

The effectiveness of the agent is achieved and demonstrated by reaching the goal as fast as possible. This made it natural to consider cost for communication. The agent requesting information is charged one step, and receives a reward of -1 for the information requested. As the agents are encouraged to be benevolent, no charge is made for giving information. In this model, an agent always replies to received request(s). A no cost for communication model is used for comparison purposes. In some environments the cost of communication is more than others. Functionality was also added to the system to test for the application of a correct policy, and a statistic was added allowing the system to track the number of consecutive correctactions.

7 Experimental Results

In this section, three RL agents systems, an autonomous agent, two independent agents, and two communicating agents, were tested. These agents are capable of only moving in four directions, requesting information, and providing information. The experiment is performed on a single autonomous agent, two independent autonomous agents, and two communicating agents. Figure 3 and 4 show the results of the independent agent. The agent was run through the maze 2000 times and several statistics were generated from the results. In Figure 3, the x-axis refers to the number of games played and y-axis refers to the number of steps required to complete the game.

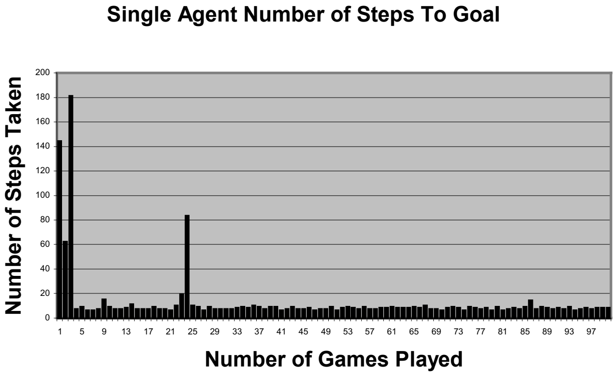


Fig. 3. Single Agent□ Number of Steps to Goal.

Figure 3 shows that many steps are taken early in the learning process, then after approximately 20 episodes the number of steps settles into a uniform distribution. Based on this and similar charts, we observed that the learning occurs in the first few episodes. Figure 4 shows the number of consecutive correct actions selected in a row. The x-axis refers to the number of correct consecutive steps taken, and the y-axis refers to the number of games played. An action is considered correct if the corresponding *Q(s, a)* value is the highest for the current state *s*. As one action is selected twice and acted on once in *SARSA*(λ), the graph values are two times the number of consecutive correct actions taken. We elected to count the number of steps in 30 episodes inclusive as a test statistic for the learning rate.

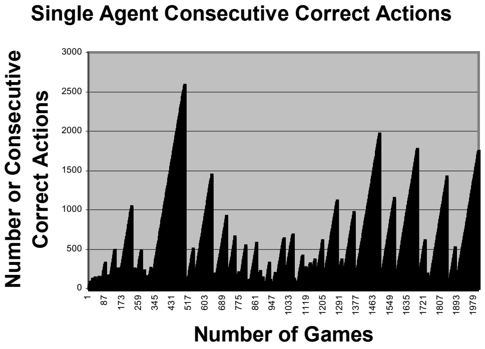


Fig. 4. Single Agent--Number of Consecutive Correct Actions.

The two statistics used to compare the agents are: the number of steps taken in the first thirty episodes, which represents the learning rate; and the average of the number of steps taken between episode thirty-one and episode 1000, which represents the learned performance of the agent(s). A range of values for *w* were tried, and we settled on a value of *w* equals to 0.501. This weights the incoming values slightly. We observed the following from the first thirty episodes:

- One autonomous agent required 698 steps.
- Two independent agents required 655 steps each.
- Two cooperative agents not charged a cost for communication required 920 steps each.
- Two cooperative agents charged a cost for communication required 1083 steps each.

We observed the following averages from episodes thirty-one to 1000:

- One autonomous agent averaged 7.88 steps per episode.
- One autonomous agent averaged 7.67 steps per episode each.
- Two communicating agents not charged a cost for communicating averaged 7.69 steps each per episode.
- Two communicating agents charged a cost for communicating averaged 7.34 steps each per episode.

8 Conclusions and Future Work

This paper described an approach that allows intelligent agents to learn while interacting with each other utilizing *SARSA*(*λ*) RL technique. The agents are designed and implemented with the capability to trade information and exchange the appropriate actions. When an agent receiving a request for a specific state from another agent, it would reply with a state action array that indicates the value of agent□s one policy for a specific state. To demonstrate the feasibility of our approach, several experiments were conducted on a maze environment. The results showed that communicating agents charged a cost for communicating take longer to learn, but perform efficiently once they have learned. We observed that the combined numbers of steps of the two agents are greater than the number of steps of a single agent, it is important to note that after learning, the two communicating agents acting in parallel do reach the goal faster than a single agent.

As future work we will study the implementation of this system in an environment more conducive to communication. Furthermore, we will improve the procedure of policy exchange. In addition the cost of communication requires further study in order to find guidelines delineating when communication is effective and to what degree.

References

[1] Bigus, J. P. and Bigus, J., Constructing Intelligent Agents Using Java, Second Edition, John Wiley and Sons, Inc., 2001.

[2] Finin, T., Labrou, Y. and Mayfield, J., "KQML as an Agent Communication Language", In Bradshaw J.M. (Ed.) Software Agents, Cambridge, MA: AAA/MIT Press, pp. 291-316, 1997.

[3] Littman, M. L., Markov Games as a Framework for Multi-agent Reinforcement Learning, in the Proceedings of the 11th International Conference on Machine Learning (ML-94)□, Morgan Kaufmann, New Brunswick, NJ, pp. 157□163, 1994.

[4] Makar, R., Magadevan S., and Ghavamzadeh M., Hierarchial Multi-Agent Reinforcement Learning, In proceedings of the Fith International Conference on Autonomus Agents, pp. 246-253. Montreal, Quebec. ACM Press, 2001.

[5] Perez-Uribe, A. Introduction to Reinforcement Learning,
Internet: http://lslwww.epfl.ch/~aperez/RL/RL.html, Accessed: October 2003.

[6] Prez-Uribe, A. and Sanchez, E. Blackjack as a Test-bed for Learning Strategies in Neural Networks, Citeseer: http://citeseer.nj.nec.com/40738.html, 1998. Accessed: October 2003.

[7] Russell, S. J. and Norvig, P., Artificial Intelligence: A Modern Approach, Second Edition, Prentice-Hall, Inc, 2003.

[8] Sen S. & Weiss G.Learning in Multiagent Systems. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* Ed. Weiss G., MIT Press, 1999.

[9] Sian S. S. Extending Learning to Multiple Agents: Issues and a Model for Muli-agent Machine learning. In Y. Kodratoff (De.), *Machine Learning – EWSL 91*. Springer-Verlag, pp. 440-456, 1991.

[10] Silver, B., Frawley, Iba, G., Vittal, J., & Bradford, K. A Framework for Multi-paradigmatic. In Proceedings of the Seventh International Conference on Machine Learning, pp. 348-358. Austin, Texas, 1991.

[11] Sutton, R. S. and Barto, A. G., Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning, Cambridge, Mass., pp. 291-312, 1998.

[12] Tan, M., Multi-agent reinforcement learning: Independent vs. cooperative learning, in M. N. Huhns & M. P. Singh, eds, Readings in Agents, Morgan Kaufmann, San Francisco, CA, USA, pp. 487□494, 1997.

[13] Weiss, G., Learning to Coordinate Actions in Multi-agent Systems. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pages 311-316, 1993.

[14] Whitehead, S. D., A complexity Analysis of Cooperative Mechanisms in Reinforcement Learning, in American Association for Artificial Intelligence□, pp. 607□613, 1991.

A Modular Architecture for a Multi-purpose Mobile Robot

Gerald Steinbauer, Gordon Fraser, Arndt M□hlenfeld, and Franz Wotawa

Institute for Software Technology, Graz University of Technology, Graz, Austria
{steinbauer, fraser, muehlenf, wotawa}@ist.tugraz.at

Abstract. In this paper we describe a design approach for mobile robots which overcomes problems of previous approaches. The problems we address are robustness, reuse and exchangeability of components, flexibility and adaptability. The approach is based on a modularization of hardware and software. Furthermore, we propose a combination of reactive behaviors with classical planning and reasoning techniques as a robust and flexible control solution. We present the modularization approach, introduce the control solution and report practical results of the design approach which is used for developing a robot platform for the RoboCup Middle Size League.

1 Introduction

The design of flexible and robust hardware and software for mobile robots is a challenging task. A good design should provide robustness, flexibility, easy exchange and reuse of components, reliable communication between components and easy adaptation of the system for new purposes. These requirements have to hold for software components as well as for hardware components. A mobile robot increases its usability in daily life if it meets these requirements

The research of mobile robot architectures has prospered within recent years. Most of the proposed architectures only faced selected software aspects, such as either robustness and flexibility by layering the software [1,2] or exchangeability, reuse and portability by modularizing the software [3]. However, hardware aspects have hardly been addressed.

To fulfill as many as possible of the above requirements, we have developed a novel design approach for mobile robots. Our approach is based on a continuous modularization of both the robot's software and its hardware.

The hardware modularization is based on an encapsulation of the robot's various physical skills into autonomous modules with defined interfaces. Therefore, hardware modules can be exchanged very simply. This allows an easy adaptation of the robot's hardware for new tasks and simple investigation of new modules or new module configurations.

The modularization of the software is based on two concepts: (1) software design and (2) software architecture. The software design provides a decomposition of functionality into layers with increasing levels of abstraction. If the robot's task and environment get more complex, purely reactive control [4] is prone to fail. Evidence

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1007-1015, 2004.

has been provided that a combination of reactive behaviors with classical AI-approaches affords a more powerful, flexible and less error-prone solution [5,6,7]. Therefore, we organize the functionality in different layers ranging from an abstract top layer with planning and reasoning capabilities down to a layer with direct hardware access.

The software-architecture deals with the implementation details. It is based on *Middleware for Cooperative Robotics* (MIRO) [3]. MIRO is a CORBA-based framework for robot applications. It is used successfully on a wide range of robot platforms. This framework provides several ready to use interfaces to sensors and actors, methods for an integration of new software modules into the framework and reliable transparent communication mechanisms between software modules. The software modules are implemented as autonomous services which interact via client/server communication. Due to the object-orientated design and the existence of defined interfaces the adaptation of the framework to our platform was quite easy. Furthermore, based on this fact an exchange of software modules within the community is possible. E.g., we extended the framework by adding interfaces to new hardware (e.g. Firewire, CAN). These extentions are now publicly available.

We use this design approach for the development of the robots that form our RoboCup Middle-Size League Team (MSL) [8]. As a result we were able to create a robot soccer team from scratch with limited human and financial resources within less than a year. Besides using the robots for soccer games we are able to use the robots for service and delivery tasks within our institute.

2 Hardware Design

In our previous studies we found four skills which are important for a mobile robot in order to fulfill a given task in a given environment, regardless whether the robot plays soccer or delivers mail within an office building. These skills are: (1) movement, (2) sensing the environment, (3) manipulating the environment and (4) information processing. These skills may differ from task to task, e.g. a kicking mechanism in a robot soccer tournament or a manipulator arm with a gripper. An encapsulation of these skills in different loosely coupled modules is the first step to a flexible hardware design. Therefore, the hardware of our robot is divided into four layers. Each layer provides one of those skills. The layers are stacked to build up the robot platform (See Figure 1).

There are no restrictions to the design of the layers themselves except that they have to provide the required skills and three predefined interfaces. A mechanical interface ensures that individual layers fit together mechanically. The fast reliable Can-Bus allows for the communication within the layers [9]; each layer can communicate directly with each other layer. A single 24 V power line provides the power supply for the layers. The Can-Bus and the power line are simply looped through the layers. Introducing new layers that provide different characteristics of a skill is easily possible by following the guidelines introduced above. Every layer is equipped with its own processing unit, either a C167 microcontroller or a Pentium-based Single-Board PC. Therefore, the individual layers are able to work autonomously.

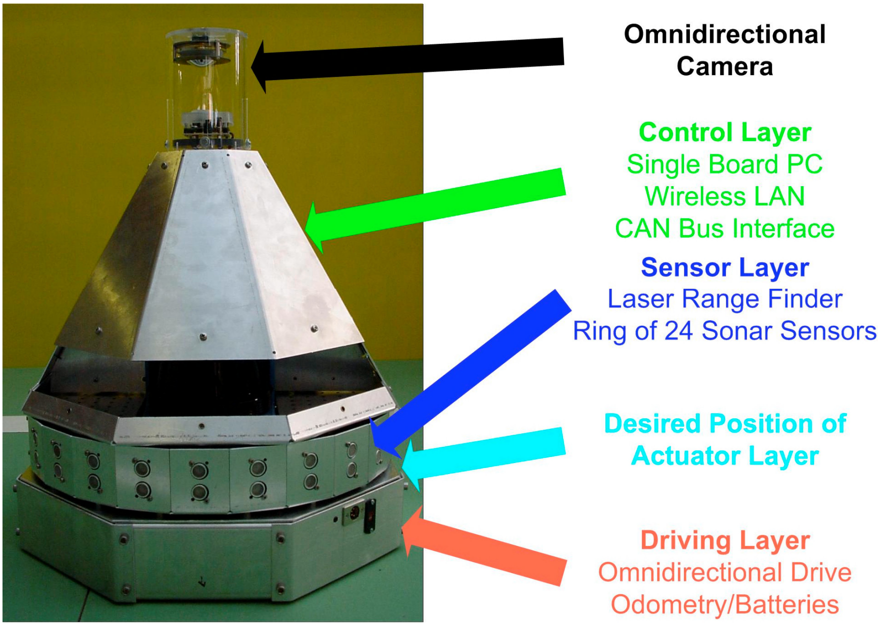


Fig. 1. The modularized Robot Platform. Actuator Layer removed.

2.1 Driving Layer

The *Driving Layer* is responsible for handling the movement of the robot. In our current design this layer is implemented as an omnidirectional drive. It is built up by four orthogonal crosswise motors each joint to an omni-wheel. By individual control of the speed and the rotating direction of each motor the robot is able to move in any direction and to rotate around its vertical axis simultaneously. This layer also hosts the battery packs to keep the center of gravity of the robot low.

2.2 Actuator Layer

All active interactions with the environment are done by the *Actuator Layer*. This layer is implemented as a pneumatic kicking device for the purpose of playing soccer games.

2.3 Sensor Layer

The *Sensor Layer* provides the entire sensing of the environment. This layer actually hosts two sensor systems. We use a Sick Laser Range Finder for proximity scans around the robot. This sensor has a high resolution (0.5°) and provides very reliable measurements. A disadvantage of this sensor is the limitation of the scan to 180°

around the robot. Therefore, this sensor is supported by another sensor system, a ring of 24 ultrasonic sensors. Those sensors have a significantly lower resolution and accuracy compared to the laser scanner but they provide a qualitative scan around the whole robot.

2.4 Control Layer

The more sophisticated information processing is done in the *Control Layer*. This includes more advanced sensory data processing, the decision making process and higher level control. Therefore, this layer is equipped with a powerful processing unit, an 850 MHz Pentium III Single Board PC with 256 MB Ram and a 20 GB hard drive. This layer also provides a communication channel to other robots or computers via a Wireless-LAN interface. Due to the field of view and the connection via the Firewire-Interface the omnidirectional camera is mounted on top of this layer.

3 Software Design

The design of the software is guided by a continuous modularization. This modularization is divided into two important aspects of the design. The first aspect deals with the functional organization of the software. It introduces a decomposition of the software in parts of similar functionality and an abstraction into layers. The second aspect deals with the logical organization of the software modules and the communication within these modules. Whereas the first aspect is important for the design and understanding of the behavior of the robot in an more abstract way, the second aspect is important for the software implementation. This distinction eases the development process due to the fact that the designer of the behavior does not have to deal with software implementation aspects and vice versa.

The idea of functional layers with different levels of abstraction is similar to the idea of cognitive robotics [10]. As we mentioned above a combination of reactive behaviors, explicit knowledge representation, planning and reasoning capabilities promises to be more flexible and robust. Furthermore, such an approach will be able to fulfill far more complex tasks. We divide the functionality of the software into three layers with an increasing level of abstraction. The functionality of a layer is based on functionality of the layer below. The layers are shown in Figure 2.

3.1 Hardware Layer

The *Hardware Layer* implements the interfaces to the sensors and actuators of the robot. This layer delivers raw continuous sensory data and performs a lowlevel controlling of the actuators. USB for the Laser Range Finder and Firewire for the omnidirectional camera are standard interfaces and already supported by our OS (Linux). The interfaces to modules on the CAN-Bus are implemented as *Virtual CAN-Connections*. A software module is able to communicate via these connections directly with one dedicated hardware module on the CAN-Bus.

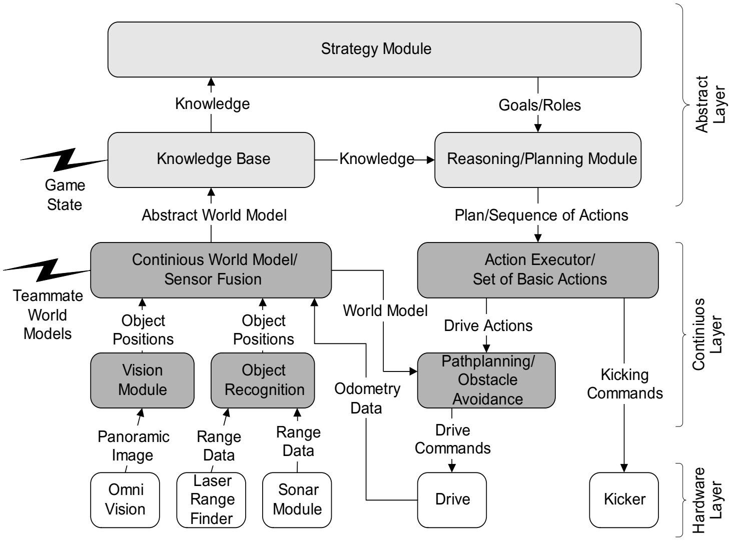


Fig. 2. Functional view of the software (Robot Soccer Example)

3.2 Continuous Layer

The *Continuous Layer* implements a numerical representation (quantitative view of the world) of the sensing and acting of the robot. This layer performs the processing of range data and mage processing. This processing provides possible positions of objects in the environment including the robots own pose. A pose consists of position and orientation of an object. Together with the motion information from the odometry these positions are fused into a continuous world model by Kalman Filters (other objects) [11] or Monte Carlo methods (own pose) [12]. For sure, all sensing and acting of a real mobile robot is afflicted with uncertainty. Therfore, sensor fusion is done using these probabilistic methods. The world model represents the continuous world by estimating the most likely hypothesis for the positions of objects. Furthermore, this layer is responsible for the execution of actions. Execution is based on a set of atomic actions implemented as simple reactive behaviors.

3.3 Abstract Layer

The *Abstract Layer* implements a symbolic representation (qualitative view of the world) about the knowledge of the robot and a planning module for the decision making. A detailed description can be found in [13]. The core of this layer is the *Knowledge Base*. It contains the entire higher-level knowledge of the robot. This knowledge consist of previously collected domain knowledge, an abstracted representation of the continuous world model and an abstract description of the actions the robot is able to perform. This knowledge is represented using a STRIPS-like representation language [14]. Based on this knowledge the *Strategy Module* chooses the next goal the robot has to achieve for fulfilling the longterm task. The *Planing Module* generates a plan (sequence of basic actions) which satisfies this goal. This plan is monitored permanently for its validity during execution. The plan is canceled or updated if preconditions for the plan are no longer valid. This plan is communicated to the Action Executor which performs the actions of the plan. The *Abstract Layer* allows for an easy implementation of a desired task by specifying the goals, actions and knowledge as logic sentences.

4 Software Architecture

The Software Architecture is based on *Middleware for Cooperative Robotics* (MIRO) [3]. MIRO is a CORBA-based framework for robot applications. It provides interfaces to a wide rage of sensors and actuators, mechanisms for simple integration of modules in the framework and a wide range of mechanisms for communication between this modules. This framework allows for a very flexible software design. The Software Architecture is shown in Figure 3.

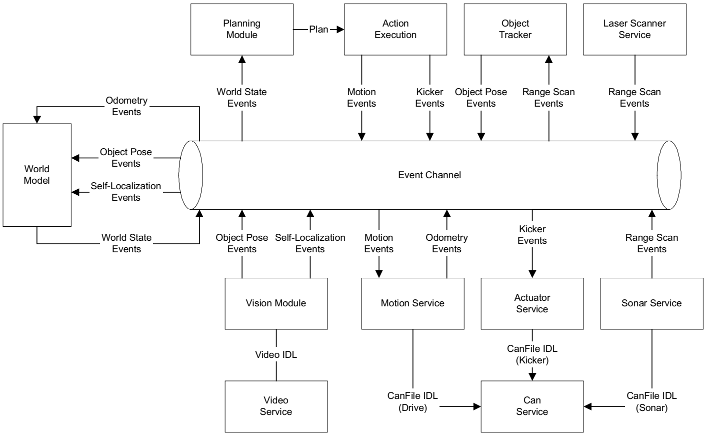


Fig. 3. Software Architecture

All software modules are implemented as autonomous services. Each service runs as an independent task. The communication between services is primarily based on two mechanisms: (1) CORBA-Interfaces and (2) Event Channel. CORBA-Interfaces are described using the Interface Definition Language (IDL) and export methods a service is able to perform. The IDL description of the interface is abstract and make no assumptions about the implementation of the interface, e.g programming language or platform. The Event Channel is a mechanism which collects and delivers events within the system. A service that produces an event simply pushes the event in the Event Channel. A service which consumes an event simply subscribes to some event type on the Event Channel. If some event of that type is available, the Event Channel delivers the event to the subscribed service. The advantage of the Event Channel is that producers and consumers do not have to be aware of each other in contrast to CORBA-Interfaces, where the client has to know the server in advance. Hence, the services are independent and an adaptation of software modules or the integration of new services is very easy and transparent. Based on the flexible design of the framework we have developed some extentions to the framework. These extensionsare mainly interfaces to new hardware we use in our robots: (1) Firewire interface for digital cameras and (2) CAN-Bus interface with virtual connections. As a matter of course we provide these extentions to the public. These extensions widen the number of platforms on which the framework could be used.

5 Practical Results

Our experiences in building a mobile robot show that this kind of design approach reduces the time and costs for developing a mobile robot and at the same time increases the flexibility and robustness of the robot. By using the design approach we were able to develop a robot soccer team from scratch with limited human and financial resources within less than a year. The basic system was implemented by a team of only nine students (32 man-months).

An other practical result shows the advantages of the modularized hardware design. We developed two versions of the Sensor Layer based on different ultrasonic sensors (Polaroid or Devantec). Both versions work transparently within the robots.

The quality and robustness of our robots were shown during the RoboCup 2003 world championship in Padova. We proceeded into the second round of the tournament at our first participation. Except damage of the Sensor Layer of one of our robots caused by an opponent, the hardware and the software ran stable without crashes. The adaptability and flexibility of our control solution were impressive during the tournament. A player's behavior could be changed easily in a few minutes on the field simply by modifying sentences that to some extent resemble human language statements.

Up to the RoboCup 2003 only few reliable information was provided by the Continuous Layer. This information was the position of ball, own goal and opponent goal, relative to the robot itself. Therefore, the knowledge about the world in the Abstract Layer was scanty. The generated plans were rather short, up to only three actions. An improvement of the sensor processing (global position of objects) in the Continuous Layer will lead to more available knowledge in the Abstract Layer and therefore to longer and more sophisticated plans.

6 Conclusion and Future Work

We presented an approach for designing autonomous mobile robots. The approach is based on a continuous modularization in software and hardware. Robot designs that are based on our approach are more flexible with regard to their intended purpose and can be easily adapted to new tasks. The hardware modularization is based on an encapsulation of skills. The software modularization is based on the MIRO-

Framework. Due to the CORBA-based architecture of the software modules an adaptation of available services and the integration of new services is very easy. The behavior of the robot itself can be easily adapted due to the different abstraction layers in the functionality and a logic based representation of knowledge, goals and actions.

We use the robot and the framework for delivery tasks within our institute to evaluate the power and flexibility of our approach. We currently work on a generalized localization module based on Monte Carlo methods which should work transparently for different environments (e.g. RoboCup MSL field or office buildings). Furthermore, we work on an improvement of the power of our knowledge representation language. We are confident that such an improvement will broaden the area of tasks where our framework could be applied. Moreover, we suggest more research should be done in plan monitoring and re-planning.

References

1. Reid Simmons, Richard Goodwin, Karen Zita Haigh, Sven Koenig, and Joseph O'Sullivan. A layered architecture for office delivery robots. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 245-252, New York, 5-8, 1997. ACM Press.
2. Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hahnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3-55, 1999.
3. Hans Utz, Stefan Sablatng, Stefan Enderle, and Gerhard K. Kraetzschmar. Miro - middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures*, 18(4):493-497, August 2002.
4. Rodney A. Brooks. Intelligence without reason. In John Myopoulos and Ray Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 569-595, Sydney, Australia, 1991. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
5. Karen Zita Haigh and Manuela M. Veloso. High-level planning and low-level execution: Towards a complete robotic agent. In W. Lewis Johnson, editor, *Proceedings of the First International Conference on Autonomous Agents*, pages 363-370, Marina del Rey, CA, 1997. (New York, NY: ACM Press).
6. D. Haehnel, W. Burgard, and G. Lakemeyer. Golex - bridging the gap between logic (GOLOG) and a real robot. In *In Proceedings of the 22nd German Conference on Artificial Intelligence (KI 98)*, 1998.
7. Miguel Arroz, Vasco Pires, and Luis Custdio. Logic based distribution decision system for a multi-robot team. In *Actas do Encontro Cientifico do Robotica 2003 - Festival Nacional de Robotica*, 2003.
8. Gerald Steinbauer, Michael Faschinger, Gordon Fraser, Arndt Mhlenfeld, Stefan Richter, Gernot Wber, and Jrgen Wolf. Mostly Harmless Team Description. In *Proceedings of the International RoboCup Symposium*, 2003.
9. K. Etschberger. *Controller Area Network (CAN) Basics, Protocols, Chips, Applications*. IXXAT Automation, 2001.
10. C. Castelpietra, A. Guidotti, L. Iocchi, D. Nardi, and R. Rosati. Design and Implemantation of Cognitive Soccer Robots. In *RoboCup 2001: Robot Soccer World Cup V*, volume 2377 of *Lecture Notes in Computer Science*. Springer, 2002.

11. M. Dietl, J.-S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01), Maui, Hawaii*, 2001.
12. Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *AAAI/IAAI*, pages 343-349, 1999.
13. Gordon Fraser. AI-Planning System for Robotic Soccer. Master's thesis, Institute for Software Technology, Graz University of Technology, 2003.
14. Richard. E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189-208, 1972.

An Agent-Based E-engineering Services Framework for Engineering Design and Optimization

Qi Hao[1], Weiming Shen[1], Seong-Whan Park[2],
Jai-Kyung Lee[2], Zhan Zhang[1], and Byung-Chun Shin[2]

[1]Integrated Manufacturing Technologies Institute
National Research Council of Canada
800 Collip Circle, London, Ontario N6G 4X8, Canada
{qi.hao, weiming.shen,zhan.zhang}@nrc.gc.ca
[2]Korea Institute of Machinery & Materials
171 Jang-dong, Yusung-Gu, Daejeon, 305-343, Korea
{swpark, jkleece, bcshin}@kimm.re.kr

Abstract. This paper presents an ongoing project on the application of intelligent software agents to engineering design and optimization. In this project, an agent-based e-engineering services framework for engineering design and optimization is proposed and being developed based on several advanced technologies, such as agents, Internet/Web, workflow and database. This framework aims at providing integrated or individual engineering services over the Internet that benefits small and medium enterprises (SMEs). A software prototype is currently being implemented using intelligent agents to integrate various engineering software tools. Scalability, intelligent load balancing and distributed decision-making are among the main characteristics that are highly expected from this system. A wheel-axle-assembly (part of a bogie system) is chosen as the test part of this prototype system.

1 Introduction

Global competition makes time-to-market the most critical concern for manufacturers. The product design, not in exception, has to adopt new technologies not only to improve its process efficiency but also to reach its targets economically.

Complex engineering design tasks generally require the cooperation of multidisciplinary design teams and the readily accessibility of various engineering tools, such as CAD, FEA, modeling, analysis, simulation and optimization packages. However, the large investments and high-demanding expertise requirements of these engineering tools hinder the small and medium enterprises from acquiring their design and optimization competencies. Even for large companies, the design process is time-consuming because of the lack of an integrated engineering environment that can coordinate and automate activities of multidisciplinary design teams. Engineering software, knowledge and design data are distributed geographically. Manual file formats transformation and delivery makes the design iterations cumbersome.

This paper will briefly introduce the project background, the system architecture of the proposed e-engineering design and optimization services framework and the pilot CRP

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1016-1022, 2004.

project between National Research Council□s Integrated Manufacturing Technologies Institute (NRC-IMTI, Canada) and Korea Institute of Machinery & Materials (KIMM, Korea). Considering it is an ongoing project, limited information will be released in this paper.

2 Project Background

KIMM is starting a long-term national R&D project on the development of □Cyber Engineering□ with initiatives to provide engineering services (including design, engineering and manufacturing technologies) to Korean industries to strengthen their competencies to win the global competition. The paradigm of □Cyber- Engineering□is shown in Fig. 1. One of two major objectives is to develop an Internet based collaborative design and optimization environment using agent technology.

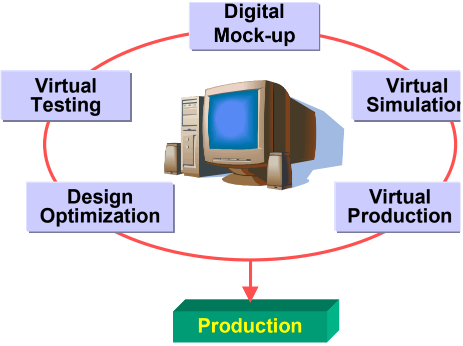


Fig. 1. Paradigm of “Cyber Engineering”

A research team at NRC-IMTI has also been working on multidisciplinary design optimization (MDO), Internet/Web/Agent based collaborative design and manufacturing environments, manufacturing systems integration and shop floor monitoring & control.

Based on the common interests, a pilot collaborative research project on the development of an e-engineering environment for design and optimization is carrying out between KIMM and NRC-IMTI early in 2003. The primary objective of this project is to develop a proof-of-concept prototype software system based on the Web and software agents for the design and optimization of a wheel-axle assembly (part of a bogie system).

3 Related Work

In agent-based collaborative design systems, intelligent software agents have mostly been used to enable cooperation among designers, to provide wrappers for integrating legacy software tools, or to allow better simulations. An earlier review of multi-agent collaborative design systems can be found in [1]. Shen et al [2] provide a detailed discussion on issues in developing agent-oriented collaborative design systems and a review of significant, related projects or systems.

The application of software agents in design has been demonstrated by a number of research projects, such as PACT [3], SHARE [4], DIDE [5], Co-Designer [6], and A-Design [7]. A recent comprehensive review of related projects can be found in [8].

NRC-IMTI project team has been working on agent-based integration in design and optimization fields for many years. A Web and agent based software environment called WebBlow [9][10] (a blow molding design and optimization environment) has been developed to integrate blow molding process simulation, performance simulation and optimization algorithms. Recently, Shen and Hao [12] extended the architecture of WebBlow to a service oriented integration framework in order to utilize engineering services over the Internet.

4 Architecture of E-engineering Services Framework for Design and Optimization

The architecture of the proposed e-engineering services framework is shown in Fig. 2.

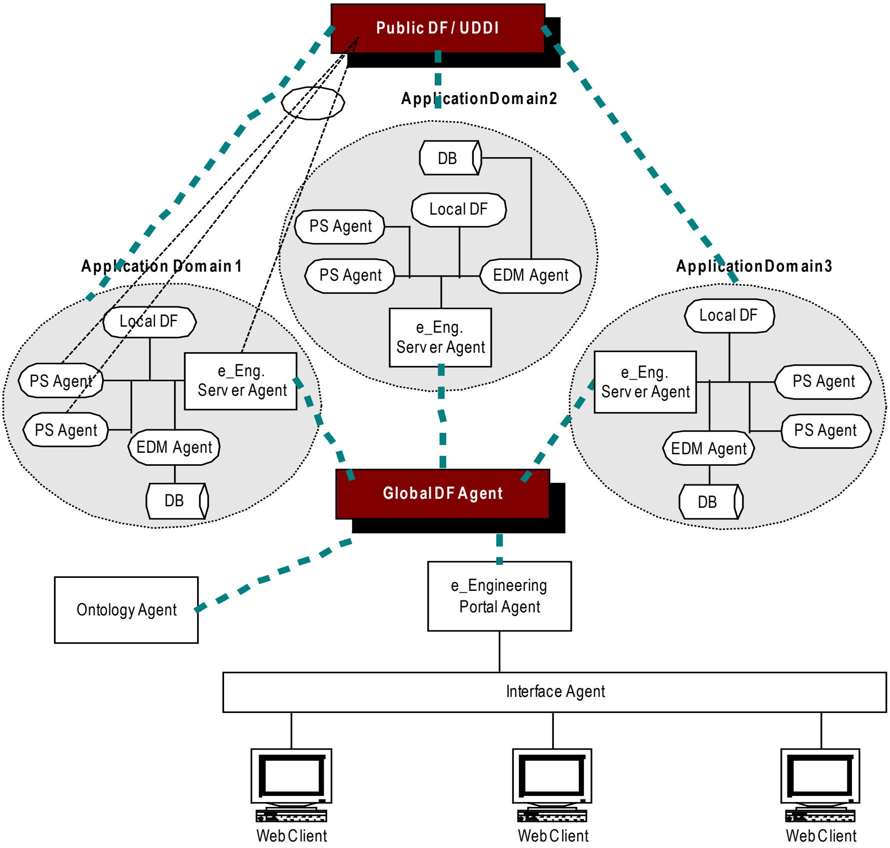


Fig. 2. Service oriented architecture for engineering design and optimization

Each application domain can be looked upon as an independent ASP (Application Service Provider), which means that each one of them is an independent problem solver and has its own expertise in design and optimization. All engineering solutions can be accessed over the Internet through the e-Engineering Portal Agent.

The characteristics of the proposed framework are summarized as follows:

- The system architecture is light weighted at the system level. Only a Global DF Agent, an integrated ASP Portal and (maybe) an ontology agent are required at this level.
- Each application domain is integrated, independent, and application oriented.
- The architecture is scalable for engineering ASP service mode. Engineering solutions can be easily plugged into the framework as individual ASPs.
- Both the entire application solution and the internal Problem Solving Agents (PS Agents) in an application domain can publish their functions and capacities to a public DF agent (UDDI) as engineering services. In this way, application domains can outsource their tasks easily.

5 Pilot Project Implementation

Based on the architecture proposed in Section 4, research and implementation plan is made for our ongoing pilot collaborative research project, as illustrated in Fig. 3.

This prototype is designed to solve the design and optimization problem of a wheel-axle assembly. The system is application oriented and can be looked upon as an individual ASP in Fig. 2. System functions include project process management, user management, engineering data management and system integration with different kinds of commercial or self-developed engineering software tools. A number of software agents have been designed. Their functions are briefly described below:

E-engineering Server Agent

It is responsible for initializing a project as a response to the project definition; generating Job Agents as a response to the workflow; tracing design jobs□status, maintaining project data and logging through the EDM Agent.

EDM Agent (Engineering Data Management)

It is a proactive database agent. Other agents may request (user or project related) information from or store information into databases or files through the EDM Agent. EDM Agent may monitor current design projects, discover the data requirements of a job, and send data files automatically to the target agents before the job starts.

Job Agent

A Job Agent communicates with EDM Agent for storing and retrieving data; with DF Agents for finding matched PS agents; and with PS Agents for negotiation based design task allocation. Job agents are created and dissolved dynamically by the e-Engineering Server Agent with the life cycle of their corresponding design jobs.

DF Agent (Directory Facilitator)
It is responsible for registering PS Agents when they become active; keeping up-to-date PS Agent information on hand; providing lookup and matching-making services to Job Agents; etc.

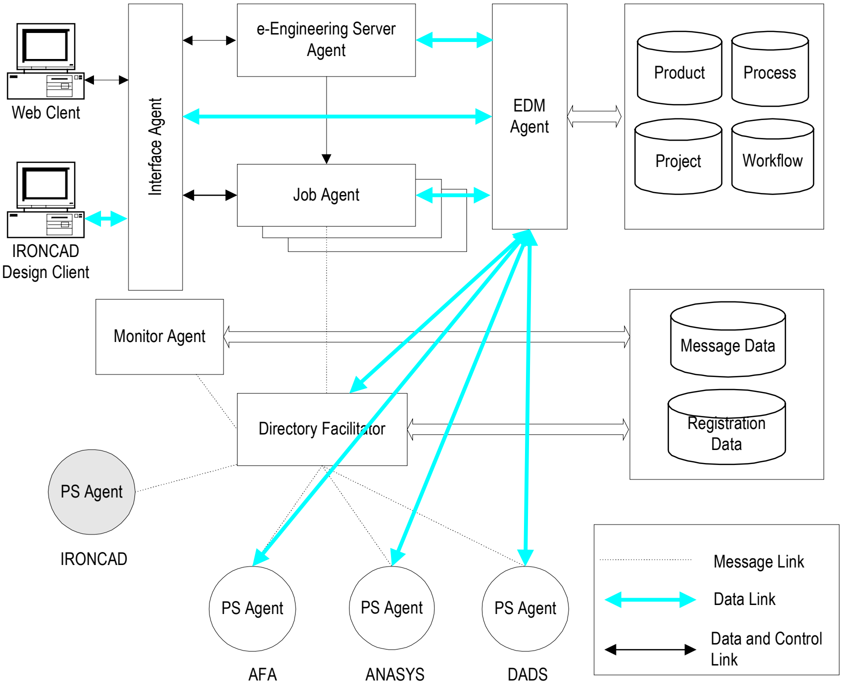


Fig. 3. Prototype for short-term pilot CRP project

PS Agent (Problem Solving Agent)
It is the actual engineering problem-solving agent. A PS agent not only carries out the communication, negotiation functions of an engineering software tool, but also executes the related analysis, simulation or optimization based on the parameters provided by EDM Agent. In this project, four engineering software tools are required to be integrated: IRONCAD v 6.0, DADS v 9.6 (for dynamic analysis), ANASYS v 7.1 (for structural analysis), and AFA (a self-developed program for fatigue analysis). Considering IRONCAD is a standalone interactive 3D modeling software tool which can only be installed on the designer□s site, its representative PS Agent is treated differently as an empty node (an agent without execution interface).

Monitor Agent
It is specially designed to facilitate the monitoring of multi-agent behaviors happened behind the user interfaces so that the system could be visible to users.

The software prototype environment is being implemented on a network of PCs with Windows 2000/XP and Linux operating environments as well as SUN workstations with the SUN UNIX environment. Java is the primary programming language for system implementation. Other programming languages including C/C++, FORTRAN, and Visual Basic, are also used for legacy systems integration. All agents are implemented on an agent framework developed at NRC-IMTI. FIPA ACL is used as the agent communication language. The EDM agent is implemented to integrate MySQL^{TM} as the primary database for the entire system. Server side modules are implemented on Apache^{TM} and Tomcat^{TM}. The prototype implementation will be completed early in 2004.

6 Conclusions

This paper presents an ongoing project between NRC-IMTI and KIMM for the development of an e-engineering environment for the design and optimization of a wheel-axle assembly. Agent, Internet/Web, workflow and database are the main technologies used in this project. Based on the (long-term) e-engineering service framework, we are currently working on the implementation of a proof-of-concept software prototype under a pilot collaborative research project. Although the prototype environment is implemented for the design and optimization of a wheel-axle system, the proposed framework can be used to implement distributed design and optimization systems for all related engineering areas including mechanical engineering and civil/architectural engineering. It can also be easily extended for applications in manufacturing and industrial engineering.

References

1. Lander, S.E.: Issues in Multi-Agent Design Systems. IEEE Expert. 12(2)(1997) 18-26
2. Shen, W., Norrie, D.H., Barth□s, J.P.: Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing. Taylor and Francis, London UK (2001)
3. Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M., Weber, J.C.: PACT: An Experiment in Integrating Concurrent Engineering Systems. IEEE Computer. 26(1)(1993) 28-37
4. Toye, G., Cutkosky, M., Leifer, L., Tenenbaum, J., Glicksman, J.: SHARE: A Methodology and Environment for Collaborative Product Development. Proc. of 2nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. (1993) 33-47
5. Shen, W., Barth□s, J.P.: An Experimental Environment for Exchanging Engineering Design Knowledge by Cognitive Agents. In: Mantyla M., Finger S., Tomiyama T. (eds.): Knowledge Intensive CAD-2. Chapman & Hall (1997) 19-38

6. Hague, M.J., Taleb-Bendiab, A.: Tool for Management of Concurrent Conceptual Engineering Design. Concurrent Engineering: Research and Applications. 6(2)(1998) 111-129
7. Campbell, M.I., Cagan, J., Kotovsky, K.: A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic Environment. Research in Engineering Design. 11(1999) 172-192
8. Wang, L., Shen, W., Xie, H., Neelamkavil, J., Pardasani, A.: Collaborative Conceptual Design: A State-of-the-Art Survey. CAD. 34(13)(2002) 981-996
9. Wang, Y., Shen, W., Ghenniwa, H.: WebBlow: A Web/Agent Based Multidisciplinary Design Optimization Environment. Computers in Industry. 52(1)(2003) 17-28
10. Shen, W., Ghenniwa, H.: A Distributed Multidisciplinary Optimization Framework: Technology Integration. Journal of Integrated Design and Process Science. 7(3)(2003) 1-14
11. Shen, W., Hao, Q.: A Service Oriented Integration Framework for Blow Molded Automotive Parts Design and Optimization, SAE 2004 World Congress. (2004) (04-CONG-8)

Robust and Adaptive Tuning of Power System Stabilizers Using Artificial Neural Networks

Farzan Rashidi[1] and Mehran Rashidi[2]

[1] Control Research Department, Engineering Research Institute, Tehran, Iran,
P.O.Box: 13445-754, Tehran
f.rashidi@ece.ut.ac.ir
[2] Hormozgan Regional Electric Co. Bandar-Abbas, Iran,
P.O.Box: 79167-95599, Bandar-Abbas
mrashidi@mehr.sharif.edu

Abstract. Tuning of power system stabilizers (PSS) over a wide range of operating conditions and load models is investigated using an artificial neural network (ANN). The neural nettwork is specially trained by an input-output set prepared by a novel approach based on genetic algorithms (GA). To enhance power system damping, it is desirable to adapt the PSS parameters in real-time based on generator operating conditions and load models. To do this, on-line measurements of generator loading conditions are chosen as the input signals to the neural network. The output of the neural network is the desired gain of the PSS that ensures the stabilization of the system for a wide range of load models connected to the power system. For training the neural network a set of operating conditions is chosen as the input. The desired output for any input is computed by simultaneous stabilization of the system over a wide range of load models using genetic algorithm. In this regard, the power system operating at a specified operating condition and various load models is treated as a finite set of plants. The problem of selecting the output parameters for every operating point which simultaneously stabilize this set of plants is converted to a simple optimization problem which is solved by a genetic algorithm and an eigenvalue-based objective function. The proposed method is applied to a test system and the validity is demonstrated through digital simulation.

1 Introduction

Low frequency oscillations are a common problem in large interconnected power systems [1]. Power system stabilizers (PSS) can provide supplementary control signal to the excitation system and/or governor system of the electric generating until to damp these oscillations and to improve generator's dynamic performance [2]. Conventional power system stabilizer is a lead-lag compensation-type device based on control theory [3] that has been adopted by most utility companies because of its simple structure, flexibility and ease of implementation, and it has made a great contribution in enhancing power system damping and dynamic stability [4]. Other types of PSS such as proportional-integral PSS [5-6] have also been proposed. The

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1023-1032, 2004.

parameters of these stabilizers are normally fixed at certain values which are determined under a particular operating condition. In daily operation of a power system, the operating condition changes as a result of load changes or unpredictable major disturbances such as a fault. Thus, a set of PSS parameters which provide good dynamic performance under a certain operating condition may no longer yield satisfactory results when there is a drastic change in operating condition. Another major drawback of the forementioned proposed designs is that the effect of load models has not been taken into account when designing PSS. Over time, the load model of power system changes, and the PSS with fixed-parameters designed for one load model can not maintain the same quality as system performance of other load models. It has become clear that assumptions regarding load model can impact predicted system performance as significantly as the models chosen for excitation systems and synchronous machines. The impact of load models on power system controls and stability limits has been demonstrated in the literature [7-9], and it has been shown that load models can have a decisive influence on power system stability and control design. To maintain good damping characteristics over a wide range of load models and operating conditions a novel approach is used in this paper based on ANN and GA. For any operating point in a selected set of grid points in the real-power/reactive-power domain simultaneous stabilization of the system over a wide range of load models is considered via a single PSS. The power system under various load models could be considered as a finite number of plants. The parameters of a PSS that can simultaneously stabilize this set of plants can be determined off-line using a GA and an objective function based on system eigenvalues. The PSS parameters computed in this manner will perform well under various load models and the stability of the system is guaranteed. These set of optimized parameters for the specified range of operating conditions is used to train an ANN. So, the gain settings of the PSS will be adapted in real-time based on on-line measurements (P,Q), using a ANN-based stabilizer. The ANN outputs (PSS gain settings) for any operating point are the optimized values ensuring system stability for the desired range of load models. To demonstrate the effectiveness of the proposed PSS, time domain simulations of a synchronous generator connected to a large power system are performed. Several different loading conditions and load models are examined. It is concluded from the simulation results that the generator can maintain good damping characteristics over a wide range of operating conditions and load models. The generator with fixed-parameters no longer yields satisfactory dynamic responses when the operating condition and/or load model is changed significantly.

2 Problem Formulation

Fig. 1 shows the system under study in which a synchronous generator is connected to an infinite bus through a transmission line. The generator is modeled by a third-order model and is equipped with static excitation and governor-turbine control systems. The voltage dependent load is connected at the generator terminal.

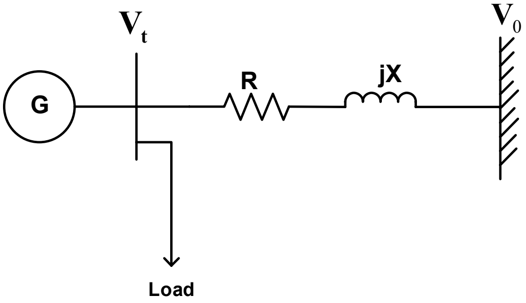


Fig. 1. Power system under study

2.1 Voltage-Dependent Load Models

This paper follows the recommendations of the IEEE Working Group [10] and many utilities [11-12] in utilizing the voltage-dependent load model for composite load representation. The load model is defined by

$$P_L = P_{L0}(V_t / V_{t0})^{np} \tag{1a}$$

$$Q_L = Q_{L0}(V_t / V_{t0})^{nq} \tag{1b}$$

Where P_L and Q_L are the load active and reactive power, n_p and n_q are load model parameters, V_t is the load bus voltage, and P_{L0}, Q_{L0} and V_{t0} are the nominal values of the load active power, load reactive power, and load bus voltage prior to a disturbance. The load representation given by Eq. (1) makes possible the modeling of all typical voltage-dependent load models by selecting appropriate values of n_p and n_q. For example, if $n_p=n_q=0$, the load is a constant-power load model, and if $n_p=n_q=2$ it is a constant-impedance load model. The measured values of the parameters n_p and n_q of various kinds f typical system composite loads (industrial, commercial, and residential) are reported in [11-12]. The system data and nominal operating point are found in [13].

2.2 Design of PI PSS

An approach based on modal control theory to design a proportional-integral (PI) power system stabilizer (PSS) for an excitation system is presented in [5-6]. The gain settings K_P and K_I of the PI PSS will be determined by left shifting the eigenvalues associated with mechanical oscillation mode of generator to the prescribed locations on the s-plane. If K_P and K_I remain fixed, the assigned eigenvalues will drift as a result of the change in system matrix A due to the changes in operating condition or load model. The excitation system with the associated PI PSS is shown in Fig. 2. Power systems experience poorly damped electromechanical oscillations due to small disturbances. These oscillations may sustain and grow if no adequate damping is available. Sustained oscillations in power systems are undesirable because they can lead to fatigue of machine shafts and to system separation. Therefore, it is desired that these oscillations be well damped. Power system stabilizer is the most widely used to enhance the damping of these oscillations. In recent years, there are a number of FACTS devices have been extensively used in providing additional damping to power system oscillations.

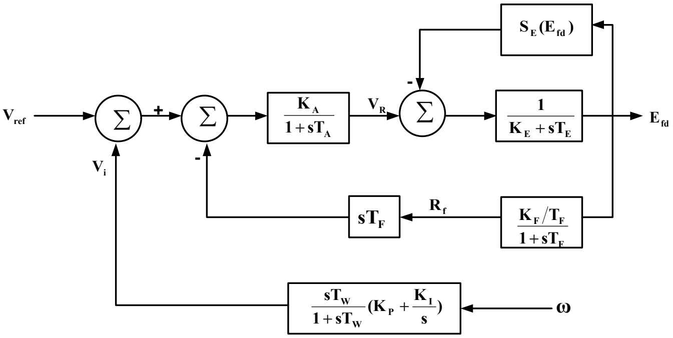


Fig. 2. Block diagram of the system with PSS

In recent years, there are a number of FACTS devices have been extensively used in providing additional damping to power system oscillations. Static Var Compensator (SVC) is one of these FACTS devices. The SVC with additional damping loop can effectively damp out these electromechanical oscillations.

3 Simultaneous Stabilization Using GAs

Genetic algorithms are global search techniques based on the operations observed in natural selection and genetics [14]. They operate on a population of current approximations (the individuals) initially drawn at random, from which improvement is sought. Individuals are encoded as strings (chromosomes) constructed over some particular alphabet, e.g. the binary alphabet {0,1} so that chromosomes values are uniquely mapped onto the decision variable domain. Once the decision variable domain representation of the current population is calculated individual performance is assumed according to the objective function which characterizes the problem to be solved. It is also possible to use the variable parameters directly to represent the chromosomes in the GA solution at the reproduction stage, a fitness value is derived from the raw individual performance measure given by the objective function and used to biasd the selection process. Highly fit individuals will have increasing opportunities to pass on genetically important material to successive generations. In this way the genetic algorithms search many points in the search space at once and yet continually narrow the focus of the search to the areas of the observed best performance. Before the ANN model can be used to provide the desired output, the model has to be trained to recognize the relationships between the input parameters (P and Q) and the related output (K_P and K_I). For any initial condition, the related PSS parameters (K_P and K_I). For satisfactory system damping depend on the load model parameters n_p and n_q. In order to have system stability over a wide range of load models and a specified operating point, it is proposed to consider the power system under various load model parameters as a finite number of plants. The parameters of a PSS that can simultaneously stabilize this set of plants can be determined off-line using a genetic algorithm and an objective function based on the system eigenvalues. Genetic algorithms are used as parameter search techniques which utilizes the genetic operators to find near optimal solutions. The advantage of the GA technique is that it is independent of the complexity of the performance index considered. It suffices to specify the objective function and to place finite bounds on the optimized parameters. Consider the problem of determining the parameters of a single PSS that simultaneously stabilizes the family of N plants (i.e., power system under various load model parameters n_p and n_q and a specified operating point)

$$\dot{X}(t) = A_k X(t) + B_k u(t) \tag{2}$$

Where $X(t) \in R^n$ is the state vector and *u(t)* is the supplementary stabilizing signal. A necessary and sufficient condition for the set of plants in equation (2) to be simultaneously stabilizable with the supplementary signal is that eigenvalues of the closed-loop system lie in the left-hand side of the complex s-plan. This condition motivates the following approach for determining the parameters (K_P and K_I) of the PSS. Select (K_P and K_I) to minimize the following objective function:

$$J = \max \mathrm{Re}(\lambda_{k,l}) \,, \mathrm{k} = 1,2,...,\mathrm{N},\ 1 = 1,...,\mathrm{n} \tag{3}$$

Where $(\lambda_{K,1})$ is the *l*th close-loop eigenvalues of the *k*th plant, subject to the constrains that $|K_P| \leq a$ and $|K_I| \leq b$ for appropriate prespecified constants *a* and *b*. clearly if a solution is found such that *J*<0, then the resulting (K_P and K_I) simultaneously stabilize the collection of plants. The existence of a solution is verified numerically by minimizing *J*. The optimization problem is easily and accurately solved using genetic algorithms. For a given operating point and a specific load model, the eigenvalues of the closed-loop system are computed and the objective function evaluated. In a typical run of the GA an initial population is randomly generated. This initial population is referred to as the zeroth generation. Each individual in the initial population has an associated objective function value. Using the objective function information, the GA then produces a new population. The application of a genetic algorithm involves repetitively performing two steps:

i. The calculation of the objective functions for each of the individuals in the current population. To do this, the system eigenvalues must be computed.

ii. The genetic algorithm then produces the next generation of individuals using the selection, crossover and mutation operators.

These two steps are repeated from generation to generation until the population has converged, producing the optimum parameters.

4 Design of the ANN for Tuning PSS

In this section, an ANN model [15] is used to tune PSS so that it can yield proper optimal gain setting for the desired range of load models under different operating points.

4.1 ANN Architectural Design

As illustrated by Fig. 3, the network is composed of many simple processing elements that are organized into a sequence of layers. These are the input layer, the hidden layer, and the output layer. In this work, generator real power output (P) and generator reactive power output (Q) are taken as the inputs of the neural network; hence two neurons are used for input in the ANN architecture. The output layer, on the other hand, consists of two output neurons representing the power system stabilizer gain settings (K_P and K_I). Between the input and output layers, generally there are at least one hidden layer. Since there is no direct and precise way of determining the number of hidden layers to use and the exact number of neurons to include in each hidden layer, one hidden layer containing eight neurons is used in this work. Research in this area [16] proved that one or two hidden layers with an adequate number of neurons is sufficient to model any solution surface of practical interest. The appropriate number of hidden neurons used is determined by experiment through the evaluation of a range of different configuration of hidden neurons. It is observed from Fig. 4 and 5 that the configuration with eight hidden neurons yields the best result.

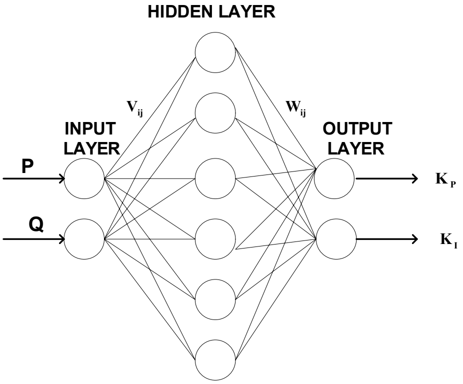


Fig. 3. The ANN model for the PSS

4.2 Data Preparation and Net Training

As already mentioned, before the ANN model can be used, it has to be trained to recognize the relationships between the input parameters and the desired outputs. These relationships will be stored as connection weights between the different neurons. The process of the determining the weight is called the training or the learning process. In order to train the ANN model to produce the desired gains, it has to be trained over the full range of typical operating points. The *P* and *Q* ranges used are between 0.1 to 1 and -0.3 to 1 respectively. Patterns within these ranges are evenly distributed so that the training can cover all possible typical load values. If this is not the case, training will tend to focus on regions where training patterns are densely clustered, and neglect those that are sparsely populated, hence producing inaccurate gains. The multilayer feed forward network used in this work is trained using the back-propagation (BP) paradigm developed in [17]. The BP algorithm uses the supervised training techniques. In this technique, the interlayer connection weights and the processing element thresholds are first initialized to small random values. The network is then presented with a set of training patterns, each consisting of an example of the problem to be solved (the input) and the desired solution to this problem (the output). The training patterns are presented repeatedly to the ANN model and weights are adjusted by small amounts that are dictated by the general delta rule [17]. This adjustment is performed after each iteration when the network's computed output is different from the desired output. This process continues until weights converge to the desired error level or the output reaches an acceptable level. The system of equations that provides a generalized description of how the learning process is performed by the BP algorithm is described by Simpson [18].

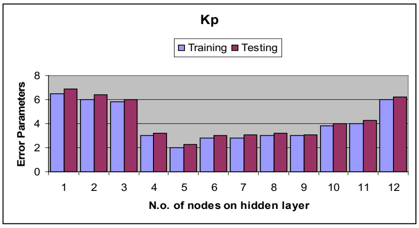


Fig. 4. Errors based on the number of hidden nodes for the gain setting K_P

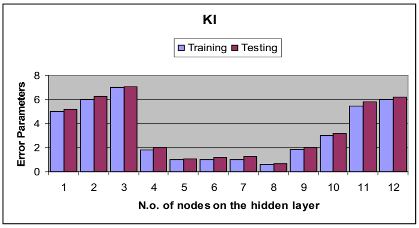


Fig. 5. Errors based on the number of hidden nodes for the gain setting K_I

4.3 Network Testing and Validation

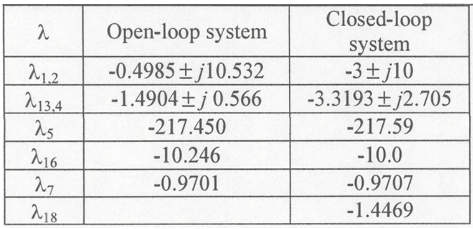
For the training set, the network prediction should be in good agreement with the actual gains. In this work, the max error was less than 1%. The generalization capability of the model should also be tested by presenting some patterns that were excluded from the data set prior to network training. In this work the max error for this case was also less than 3%. More details about the test results are presented in the next section.

5 Results and Tests

To better understand the need for stabilization, the system is first analyzed without any supplementary signals (u=0). The eigenvalues of the system without PSS (open-loop system) under constant impedance load model *(n_p=n_q=2)* and nominal operating point (P_g=1 and Q_g=0.69) are listed in the first column of Table 1. The first pair of complex-conjugate eigenvalues $\lambda_{1,2}$ are associated with the mechanical mode of oscillation of the generator. It can be seen that the damping for this oscillation mode (real parts of $\lambda_{1,2}$) is not adequate.

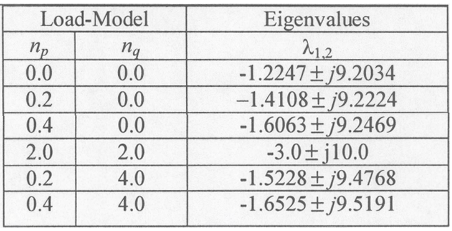
The poor damping of this mode can also be seen from the system response shown in Fig. 6. The damping of this oscillation mode needs to be improved by the proportional-integral (PI) PSS already explained. If the pair of mechanical mode eigenvalues $\lambda_{1,2}$=-3 ± j10, are selected as the desired locations, then the gain settings K_P and K_I can be computed as K_P=*11.4* and K_I =-23.85. The eigenvalues of the closed-loop system (system with PI PSS) are shown in the second column in table 1. It is found that the associated mechanical-mode eigenvalues $\lambda_{1,2}$ are exactly assigned. If the operating point remains the same, for the fixed PSS gains the mechanical-mode eigenvalues will drift when there is a change in load model. Table 2 gives the mechanical-mode eigenvalues with the PSS designed at constant impedance model under different load model. Considerable movements in these eigenvalues are occurred.

Table 1. System eigenvalues under constant impedance load mode and nominal operating point



λ	Open-loop system	Closed-loop system
$\lambda_{1,2}$	-0.4985 ± j10.532	-3 ± j10
$\lambda_{13,4}$	-1.4904 ± j 0.566	-3.3193 ± j2.705
λ_5	-217.450	-217.59
λ_{16}	-10.246	-10.0
λ_7	-0.9701	-0.9707
λ_{18}		-1.4469

Table 2. Mechanical-mode eigenvalues with the fixed PI PSS at different load models



Load-Model		Eigenvalues
n_p	n_q	$\lambda_{1,2}$
0.0	0.0	-1.2247 ± j9.2034
0.2	0.0	–1.4108 ± j9.2224
0.4	0.0	-1.6063 ± j9.2469
2.0	2.0	-3.0 ± j10.0
0.2	4.0	-1.5228 ± j9.4768
0.4	4.0	-1.6525 ± j9.5191

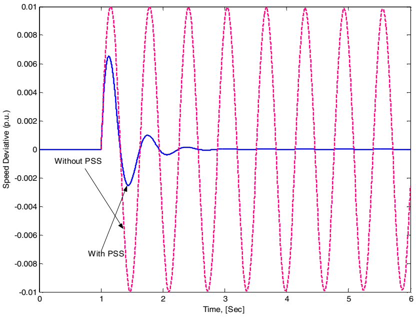


Fig. 6. The system response at n_p=n_q=*2* for 5% change in input mechanical torque

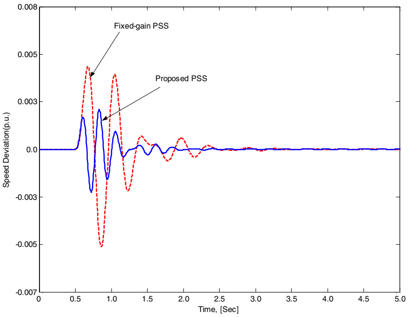


Fig 7. The system response at n_p=*1.2* and n_q=*1.9* and nominal operating point

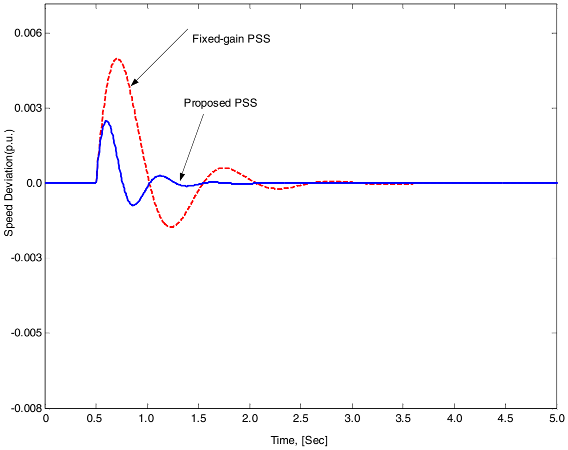


Fig. 8. System response at n_p*=2.3* and n_q*=0.9* and P=0.8 and Q=0.5

From these figures, it can be seen that the system with the fixed-gain PSS designed at one load model and operating point becomes unsatisfactory under another load model and/or operating condition, while the system is well damped with the proposed PSS. These results clearly demonstrate the capability of the artificial-intelligence for tuning PSS over the desired conditions.

6 Conclusion

An artificial neural network (ANN) has been developed for tuning a power system stabilizer. The ANN receives operating point parameters P and Q provide the desired PSS gain settings K_P and K_I. These output parameters are the optimal values that ensure system stability for the input condition and a wide range of load models. Prior to the training process a training data set consisting of a full range of typical operating points and the desired PSS gains are first prepared by using genetic algorithms. The power system operating at various load models and a specified operating point is treated as a finite set of plants. The problem of selecting the parameters of a power system stabilizer which simultaneously stabilize this set of plants has been converted to a simple optimization problem, solved by a genetic algorithm and an eigenvalue-based objective function. Hence, this process is repeated for the selected set of grid points in the operating points domain and the desired parameters are used to train the ANN until good agreement between predicted gains and the actual gains is reached. Once the ANN is adequately trained, the network is then tested to ensure that it can appropriately predict the correct gains given operating point parameters that are not included in the training data set. Simulation results show that when the gain settings of the PSS are updated in real time by the ANN. The PSS can provider good damping for the power system over a wide range of operating conditions. In addition, the gain settings can provide acceptable damping for a wide range of load models considering any operating point. This capability is due to the application of genetic algorithms as a powerful optimization tool. On the other hand, a PSS with fixed-gain settings can only provide a good damping effect under some particular load model and/or operating condition.

References

1. Yao-Nan Yu " Electric power system dynamics", new york Academic Press 1983
2. F. P. Demello and T. F. Laskowski, " Concepts of power system dynamic stability", IEEE Trans. On power Apparatus and System, Vol. PAS-94, No. 3, pp. 827-833, may/June 1975
3. F. P. DeMello and C. A. Concordia, " Concepts of synchronous machine stability as affected by excitation control", IEEE Trans. On power Apparatus and system, Vol. PAS-98, No. 4, pp. 316-329, April 1969
4. E. V. Larsen and D. A. Swann "Applying power system stabilizers: part 1-3", IEEE Trans. On power Apparatus and system, Vol. PAS-100, No. 6, pp. 3017-3046, June 1981
5. Y. Y. Hsu and C. Y. Hsu, " Design of a proportional-integral power system stabilizer" IEEE Trans. On power system, Vol. 1, No. 2, pp. 46-53, 1986
6. Y. Y. Hsu and C. C. Su, " Application of power system stabilizer on a system with pumped storage plant" IEEE Transactions on power system Vol. 2, pp. 80-86, 1988
7. K. A. Ellithy and M. A. Choudhry, " Effect of load models on AC/DC system stability and modulation control design", IEEE Transactions on power systems, Vol. 4, pp. 411-418, 1989
8. E. Vaahedi, H. M. Zein El-Din and W. W. Price, "Dynamic load modeling in large scale stability studies", IEEE Trans. On power system, Vol. 3, pp. 1039-1045, 1988
9. W. Mauricio and A. Semlyen, "Effect of load characteristics on dynamic stability of power systems", IEEE Trans. On power Apparatus and systems, Vol. PAS-91, pp. 2295-2304, 1972.
10. IEEE computer Analysis of power system working Group, System load dynamics-Simulation effects and determination of load constants, IEEE Trans. Power Apparatus and systems, Vol. PAS-92, pp. 600-609, 1973.
11. T. Ohyama , A. Watanabe, K. Nishimura and S. Tsurata " Voltage dependence of load composite loads in power systems", IEEE Trans. Power Apparatus and systems, Vol. PAS-104, pp. 3064-3073, 1985
12. Wen-shiow Kao, C. J. Lin and C. T. Huang, " Comparison of simulated power system dynamics applying various load models with actual recorded data", IEEE Trans. On power Systems, Vol. 1, pp. 248-254, 1994
13. P. M. Anderson, A. A. Fouad, "power system Control and stability", The lowa Univ. Press, Ames, lowa, 1990
14. D. E. Goldberg, " Genetic algorithms in search, optimization and machine learning", Addison-Wesley, Reading, 1989.
15. J. Stanley, " Introduction to neural networks", California Scientific Software, Siera Madre, CA, 3rd edition, 1990
16. A. Lapedes and R. Farber, "How neural networks work", Neural Information Processing Systems, American Institute of Physics, New York, pp. 442-456, 1988.
17. D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing: Exploration in the Microstructure of Cognition", Vol. 1, Foundations, MIT Press, Cambridge, MA, 1986
18. P. K. Simpson, "Artificial neural systems: Foundations, Paradigms, Applications, and Implementations", Pergamon, Elmsford, NY, 1990.

Modified Bifurcating Neuron with Leaky-Integrate-and-Fire Model

Lon Risinger and Khosrow Kaikhah

Department of Computer Science
Texas State University
San Marcos, Texas 78666
lr1036@TxState.edu
kk02@TxState.edu

Abstract. The Modified Bifurcating Neuron (MBN) is a neuron model that is capable of amplitude-to-phase conversion and volume-holographic memory. Inputs are real valued and temporally spaced. This allows information to be coded in the temporal spacing of inputs and outputs as well as their values. At its core, the MBN incorporates a stateful leaky-integrate-and–fire neuron model. The MBN attempts to produce these properties by simulating mechanisms present in biological neural systems to a greater extent than is normally found in artificial neural networks. MBNs use an object model rather than the normal linear algebra approach. The MBN is conceptually based on the computational model presented in the "Bifurcating Neuron Network 2" by G. Lee and N. Farhat

1 Introduction

The MNB is conceptually based on the Bifurcating Neuron (BN) [1] is a neuron model in which an integrate-and-fire neuron is augmented by coherent modulation from the neural environment. The BN is capable of amplitude to phase conversion and volume-holographic memory. Because of its integrate-and-fire activation model, it exhibits frequency response to incoming pulse timing. When used in a network, BNs have time delays between neuron connections that represent signal propagation latency [1]. A single BN is defined by the following three equations:

$$\theta_i(t) = 1 \tag{1}$$

$$\frac{dx(t)_i}{dt} = c_i \tag{2}$$

$$\rho_i(t) = \rho_0 \sin 2\pi ft \tag{3}$$

where $\theta_i(t)$, and $\rho(t)_i$ are the threshold level, and the relaxation level of BN$_i$, respectively. The potential $x_i(t)$ rises at a constant rate, c_i, due to the incoherent signal, until it reaches the threshold level θ_i. Then the internal potential immediately drops to the relaxation level $\rho(t)_i$. The threshold level is constant. The relaxation

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1033-1042, 2004.

level, Eq (3) is driven by the coherent signal and maintains a sinusoidal oscillation with maximum amplitude ρ_0 and frequency *f*. See Figure 1. [1]

The MBN takes a simulation or mimetic approach and loosely attempts to use structures found in biological systems rather than pure mathematical solutions. This allows us to explore the role of some biological occurrences, as they pertain to neural computation. For instance, certain behaviors or rather behavioral modes have been correlated to the presence of Theta rhythms in the mammalian brain [1], [2]. MBNs are modeled using an object oriented paradigm and not the traditional linear algebra approach. The behavioral complexity of MBN processing elements is more complex than is normally found in neural networks.

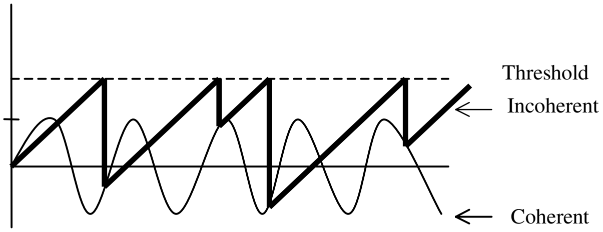


Fig. 1. Firing behavior of BN

2 Modified Bifurcating Neuron Object Model

Our Modified Bifurcating Neuron (MBN) attempts to approach the afore mentioned behavior of the BN using simulated biological mechanisms. Its structure is depicted in Figure 2.

The neuron object is where most of the processing occurs. Information propagated by each neuron is modeled by pulse objects. Each pulse corresponds to a spike in a spike train i.e. neuron output. Each neuron contains an axon and a collection of dendrites. These objects serve as containers for incoming and outgoing pulses. The neuron object delegates direct control of pulse objects to these containers. Connections between neurons are managed by synapse objects. Synapses maintain a collection of pulses, which are waiting to arrive at the neuron in question, and contain information about the time delay between neurons and the connection strength for this particular connection. A synapse can be between at most two neuron objects. Time is defined in terms of a universal clock tick that all objects receive. When pulses that have been scheduled to arrive at a neuron held in a synapse actually arrive, they contribute to the internal potential of a neuron object. If this neuron reaches its threshold, a new pulse is generated and sent to every synapse connected to the axon of this neuron.

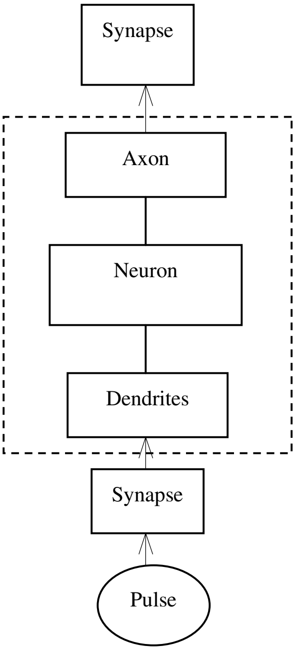


Fig. 2. MBN Object Model

3 Modified Bifurcating Neuron Networks

MBNs are linked using synapse objects. These connections are one-way and may be redundant allowing the construction of high order networks. Pulse objects propagate along these connections. In real time they are held by synapse objects until their scheduled arrival time, at which time they contribute to the rise in potential of the neuron in question.

4 Modified Bifurcating Neuron Definition

We altered the basic BN definition to model our Modified Bifurcating Neuron (MBN). The form of the BN definition has been maintained to illustrate similarities and differences between the two models. Foremost, we see that the change in internal potential, Eq (5), is no longer constant and the resting potential, Eq (6), no longer oscillates. The set of equations to follow describe the behavior of an MBN, augmented with a coherent and incoherent signal input, without an actual data input. They describe the behavior of the i^{th} MBN, which we shall simply refer to as MBN$_i$:

$$\theta_i(t) = K_i \tag{4}$$

$$\frac{dx(t)_i}{dt} = \frac{d\psi(t)_i}{dt} + \frac{d\phi(t)_i}{dt} + \beta\frac{dx(t-1)_i}{dt} \tag{5}$$

$$\rho_i(t) = C_i \tag{6}$$

where $x_i(t)$, $\theta(t)_i$, and $\rho(t)_i$ are the internal potential, the threshold level, and the relaxation level of MBN$_i$, respectively. Contributions to the internal potential of MBN$_i$ noted in Eq (5) come from the incoherent signal, $\psi(t)_i$, and coherent signal, $\phi(t)_i$. The potential $x_i(t)$ changes in time due to contributions from the incoherent signal, the coherent signal, and the potential remaining from the previous time step. This remaining potential is modified by the 'leak' factor, β (a constant between 0 and 1) representing decay. This continues until the potential reaches the threshold level $\theta_i(t)$, which is some constant value K. The internal potential then immediately drops to the relaxation level $\rho_i(t)$, which is a constant value C$_i$. Eq (5) is a recursive function dependent on a discrete time step.

4.1 Incoherent Signal

An incoherent signal is provided by a regular pulse of variable frequency. In an MBN network, this can be produced by an MBN that is connected to every other MBN and itself. This MBN produces a pulse on a constant time interval. This pulse arrives at every other MBN simultaneously and serves to increase the neuron potential as in Figure 1. The rhythmic pulse can be generated utilizing one recurrent connection, whose time delay serves to regulate the firing frequency. This is intended to be loosely synonymous with rhythmic pulses emanating from the hypocambal area in mammalian brains. Various studies have shown a correlation between frequency ranges of this rhythmic signal and general behavioral states in mammals such as heightened alertness, concentration and problem solving, hypnosis, sleep, etc. [3]. Different frequency ranges have been given different names such as Theta (7 – 10 Hz) and Gamma (60 – 100 Hz) [4]. Regardless of its specific role or roles in behavioral states, it is sufficient to say that this rhythmic pulse exists, is dynamic, and is propagated to various areas of the mammalian brain.

4.2 Coherent Signal

The MBN coherent signal is not specifically sinusoidal as in the BN. It is a spike train, whose pattern repeats over a certain time interval. In other words, it is a series of pulses of various temporal spacing and possibly various amplitudes, which repeat over a time period. It can be spaced in time in such a way that it produces an approximate sinusoidal response in the internal potential of an MBN, but it is sufficient that this signal causes a change in the internal potential that is not constant overtime within the period, Eq (5). In Figure 3, we see a theoretical depiction of an MBN stimulated by coherent signal alone. We can see that the presence of the coherent signal affects the system in a similar way as in the BN (Figure 1). It causes changes in the timing of threshold events i.e. neuron activation. This couples the temporal spacing of the output spike train to that of the coherent signal.

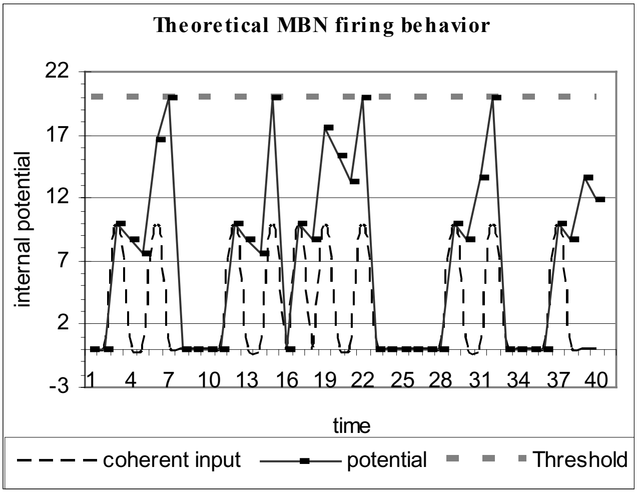


Fig. 3. Theoretical MBN firing behavior: The internal potential is driven to the firing threshold by simulated Coherent input in this example. This figure illustrates coherent input of one period or less, therefore behavior arising from periodic behavior is not illustrated here. This figure is reminiscent of Figure 1 and illustrates how input leads to temporal spacing action potential generation. Additionally we see that internal potential falls to zero after a threshold event.

4.3 MBN Behaviors

The MBN is very similar to the BN in many respects. It exhibits similar behaviors, namely amplitude to phase conversion. However, the mechanisms by which these behaviors are achieved are dramatically different. The BN represents a neuron augmented by input from the neural environment. Therefore, its coherent input is represented as an internally generated sinusoidal wave that controls the neuron resting potential, whose frequency is supplied as a network input. The MBN does not seek to make this assumption but to an extent attempts to simulate the environment as well. The entire coherent signal is supplied externally as network input. The form of this input corresponds with neuron output so that output from one or more neurons could serve as coherent input for others. This adds flexibility to the use of this signal. Furthermore on a neuron level, the coherent input is not different from other inputs. More importantly, it contributes directly to the internal potential and has no direct effect on the resting potential. In this same vein, rise in potential due to the incoherent signal is no longer constant but quasi-constant and supplied by a rhythmic pulse. Also, the internal potential of the MBN dissipates with time following the leaky-integrate-and-fire model, while the BN uses an integrate-and-fire model. The major differences between the BN and MBN are: 1) In the MBN, the coherent signal also contributes to the internal potential rise, where it does not in the BN. 2) The potential rise due to the incoherent signal is no longer constant, Eq (5), but merely quasi-constant. 3) The internal potential dissipates with time. This follows the leaky capacitor integrate-and-fire neuron model, where the internal potential of a given processing element decreases over time. Therefore, the contribution to the internal potential of a given processing element from two pulses will not be additive unless the arrival of these pulses are closely spaced in time.

4.4 General Form

It is important to remember that the MBN need not be augmented by incoherent and coherent inputs. Coherent and incoherent refer to a particular structure of the inputs. While this structure can result in rich and interesting or useful behavior, all inputs in general are handled in the same way. Thus we can construct a general form that makes no distinction between different types of network input. A single MBN is defined by the following three equations:

$$\theta_i(t) = K_i \tag{7}$$

$$\frac{dx(t)_i}{dt} = \frac{d\lambda(t)_i}{dt} + \beta \frac{dx(t-1)_i}{dt} \tag{8}$$

$$\rho_i(t) = C_i \tag{9}$$

where $x_i(t)$, $\theta(t)_i$, and $\rho(t)_i$ are the internal potential, the threshold level, and the relaxation level of MBN$_i$, respectively. Here we make no distinctions as to the structure of the input, $\lambda(t)_i$. The potential $x_i(t)$ changes in time due to contributions from an input, $\lambda(t)_i$, and the potential 'leak', β, from the previous time step. An interesting behavior emerges in Eq (8), if we consider the input as a growth rate and the 'leak' as a decay rate, where the decay rate is an exponential of the form $\beta = \alpha e^{-\alpha}$. We can see that when the growth rate equals the decay rate, the rate of change of the internal potential is zero. This implies that at some point an equilibrium condition will arise in which the internal potential will stabilize at or about some value. In general, it seems that there are two cases to consider: First, the case in which the growth rate is constant in time. In this case it can be shown, that the internal potential will asymptotically approach some static equilibrium point, if we cast, Eq (8) into a continuous form. Second, the case in which the growth is not constant in time. In this case, no static equilibrium exists. However, if we assume the growth rate periodically oscillates around some median value, it can be shown that the internal potential will reach an oscillatory equilibrium about some other median value. We see that this oscillation of the growth rate can be said to drive the internal potential in a steady state. Thus we see a theoretical basis for making distinctions in the structure of the input. This approach to a stable state may also be construed as adaptive behavior, i.e. neural plasticity. It somewhat mimics the ability of biological neurons to adapt to sustained input.

5 Expected Outcome

We now examine the expected behavior of the MBN. First consider the MBN driven by the incoherent signal alone. The incoherent signal is synonymous with a signal that is constant in time. However, due to the discreet nature of the pulse data, this cannot be achieved. Therefore our incoherent signal is only quasi-constant in time. This is represented as a rhythmic pulse of constant amplitude and frequency. According to our discussion in section 4.4, the internal potential will approach a constant value in which the growth rate equals the decay rate given, a constant input. Therefore, we would expect our quasi-constant input to approach this behavior. However, according to section 4.4, a periodic input will result in a periodic oscillation of the potential.

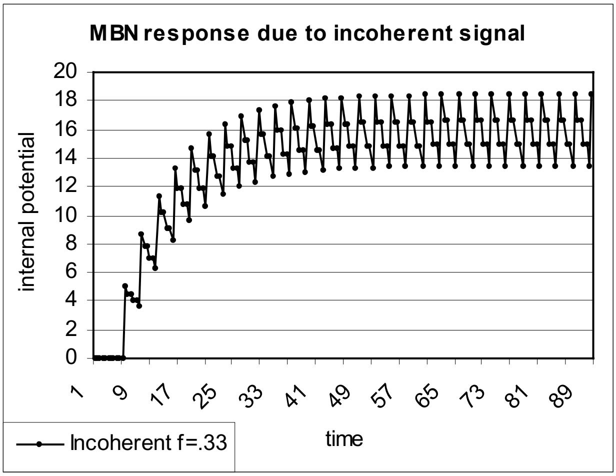


Fig. 4. MBN response due to incoherent signal: An MBN is driven by a quasi-constant input of frequency 0.33 Hz. The internal potential is seen to oscillate periodically around some median value, which asymptotically approaches an equilibrium constant.

Figure 4 depicts a sample output of an MBN supplied with this incoherent input. The result is as expected. The internal potential approaches an oscillatory equilibrium in which it oscillates predictably about some median value that asymptotically approaches a constant value. Likewise for the coherent input, we would expect a similar response. Figure 5 depicts the response of an MBN to coherent input. In this instance, coherent input is titanic, which means that the signal is composed of a series of pulse inputs closely spaced in time (one time step) followed by a period of no input. The whole cycle is then repeated. While this is not sinusoidal, it fits our requirement that the signal is not constant in time and is periodic. The fact that a titanic signal is acceptable here is interesting, considering the prevalence of titanic signals in biological systems.

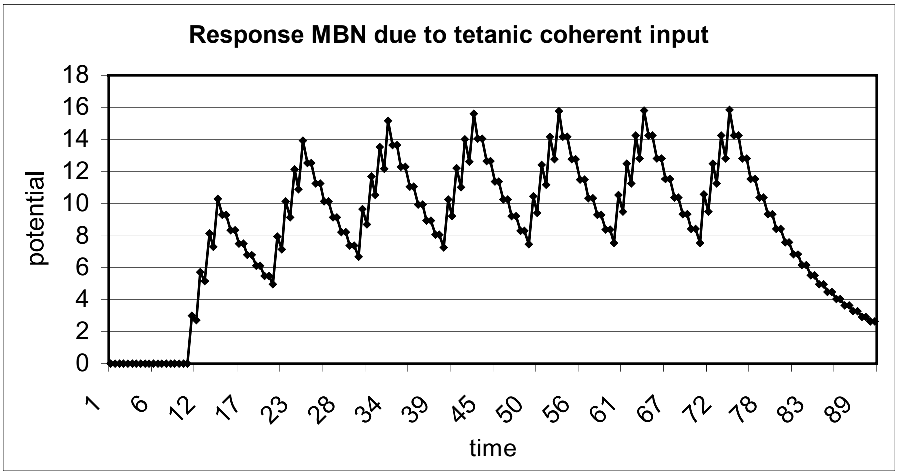


Fig. 5. Response of MBN due to titanic coherent input: Titanic coherent input consists of multiple closely spaced pulses (titanic), which occur periodically. The internal potential is seen to oscillate periodically around some median value. The trailing tail marks the conclusion of the input signal. The potential then falls off exponentially.

5.1 Amplitude to Phase Conversion

In Amplitude to phase conversion, we start with input patterns that contain both incoherent and coherent signals. These are arranged in such a way that the MBN produces output spikes on regular intervals that correspond to the period of the coherent input. If an additional input or inputs are introduced during a coherent input period, this causes a phase lead in the output spike for that period. In other words, the MBN fires before its normal firing time by some amount proportional to the amplitude of the additional input signal. If this signal is repeated in the next period this phase lead will appear again. This effect can be summarized with the addition of a new term to Eq (5), representing the contribution to the internal potential from the additional input denoted by ν(*t*), referred to as the data input.

$$\frac{dx(t)_i}{dt} = \frac{d\nu(t)}{dt} + \frac{d\psi(t)_i}{dt} + \frac{d\phi(t)_i}{dt} - \beta\frac{dx(t-1)_i}{dt} \quad (10)$$

Figure 6 depicts an example of the MBN response due to incoherent and coherent input compared to the response of incoherent, coherent, and data input signals. The output pulse timing of response including the data signal is phase shifted.

5.2 Holographic Paging

Now consider a more general case where the contributions to the internal potential from the data section at time t are given by ν(*t*). If ν(*t*) is maintained, but the coherent signal, $\phi(t)_i$, is altered, it is apparent that the rate of increase of the internal potential and thus the time spacing of output spikes will be altered. In fact, changing the coherent signal should change intervals on which the MBN fires even without a data input. Thus, the output spike pattern will only be the same, if both the coherent signal and the data input pattern are unaltered. Significant changes in either will result in an output pattern that does not match the original. Thus a particular output pattern can only be recreated given the appropriate coherent signal. The second major requirement is that storing a new data pattern with a different coherent input does not disturb the original pattern. The original BN accomplished this in a nearest neighbor pulse coupled neural network PCNN, using higher order synaptic connections. This requires that one or more new connections be added for each stored pattern. The time delays of new synaptic connection are offset by an amount proportional to the induced phase shift.

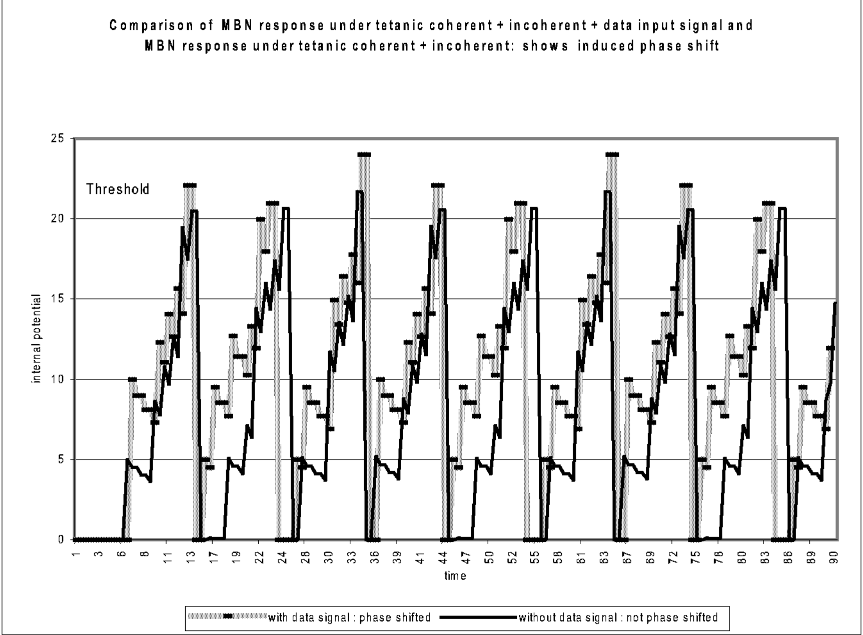


Fig. 6. Response of MBN due to titanic coherent input: Input containing a data signal as well as coherent and incoherent input is phase shifted with respect to the same coherent and incoherent input without the data signal. The firing threshold is twenty. The firing time phase shift is not constant, but this pattern of irregularity repeats over several firing intervals. Notice that the phase shift of the firing time at t = 23 is the same as the spike at t = 54 and t = 85. This is due to the discreet nature of the time sampling

6 Conclusions

The MBN more closely mimics biological neural systems than conventional neural models. MBNs exhibit diverse temporal behavior. The MBN responds differently to different structured inputs. Using these strategies, we have demonstrated amplitude to phase conversion and a theoretical basis for holographic memory. Furthermore the MBN is capable of higher order recurrent connections. This allows construction of complex temporal patterns. As stated in BNN 2 by Lee and Farhat [1], these complex temporal patterns are possible examples of how and why the brain uses multiple recurring connections. There are approximately 10^4 recurrent connections in the brain for each neuron. The reduction of these connections to one seems to be an oversimplification.

Neural Networks, in general, have proven to be powerful computational tools. Perhaps borrowing more behaviors from biological systems can lead to even more powerful tools and a better understanding of the systems that inspired them.

References

1. G. Lee and N. Farhat, The Bifurcating Neuron Network 2: an analog associative memory. *Neural Networks* 15 (2002), p. 60-84.
2. D. Rizzutto and M. Kahana, An Autoassociative Neural Network Model of Paired-Associative Learning, *Neural Computation* 13 (2001), p.2075-2092.
3. B. Good, The Theta Rhythm, (1997).
4. M. Olufsen, M. Whittington, et al, New Roles for Gamma Rhythm:Population Tunning and Preprocessing for the Beta Rhythm, *Journal of Computational Neuroscience* 14 (2003), p. 33-54.
5. C. Koch, *Biophysics of Computation: Information processing in Single Neurons*, Oxford University Press, 1999.
6. M.F. Bear, B.W. Connors, M.A. Paradise, *Neuroscience: Exploring the Brain*, Sec Ed, Lippincott Williams & Wilkins, Maryland 2001.
7. E.R. Kandel, J. H. Schwartz, T.M. Jessel, *Essentials of Neuroscience and Behavior*, Appleton & Lange, Connecticut 1995.
8. M. Eldefrawy and N. Farhat, The Bifurcating Neuron: Characterization and Dynamics In *Photonics for Computers, Neural Networks and Memories*, SPIE proceedings, San Diego, CA, (July 2, 1992), vol 1773 p.23-34.

An Application of Elman's Recurrent Neural Networks to Harmonic Detection

Fevzullah Temurtas[1], Rustu Gunturkun[2], Nejat Yumusak[1], Hasan Temurtas[2],
and Abdurrahman Unsal[2]

[1] Sakarya University, Department of Computer Engineering, Adapazari, Turkey
[2] Dumlupınar University, Department of Electric - Electronic Engineering, Kutahya, Turkey

Abstract. In this study, the method to apply the Elman's recurrent neural networks for harmonic detection process in active filter is proposed. The feed forward neural networks were also used for comparison. We simulated the distorted wave including 5th, 7th, 11th, 13th harmonics and used them for training of the neural networks. The distorted wave including up to 25th harmonics were prepared for testing of the neural networks. Elman's recurrent and feed forward neural networks were used to recognize each harmonic. The results show that these neural networks are applicable to detect each harmonic effectively.

1 Introduction

Power quality has received increased attention in recent years with the widespread application of nonlinear loads employing advanced solid-state power switching devices in a multitude of industrial and commercial applications. The operation of solid-state power switching devices in power electronic converters deteriorates the power quality by injecting harmonics into the power system causing increased distortions, equipment and load malfunctions and losses [1-3].

AC power systems have a substantial number of large harmonic generating devices, e.g. adjustable speed drives for motor control and switch-mode power supplies used in a variety of electronic devices such as computers, copiers, fax machines, etc. These devices draw non-sinusoidal load currents consisting primarily of lower-order 5th, 7th, 11th, and 13th harmonics that distort the system power quality. [3]. With the widespread use of harmonic-generating devices, the control of harmonic currents to maintain a high level of power quality is becoming increasingly important. Harmonic standards (e.g. IEEE 519 and IEC 555) have been developed to address limits in allowable harmonics [4].

A common remedial measure for reducing the effects of harmonics is passive filtering [5]. The addition of passive "LC" filters alters, or interferes, with the system impedance, and is known to cause resonance with other network impedances and can result in an excessive amplification of harmonics rather than harmonic reduction. In addition, passive filters cannot adapt to changing harmonic generating loads, thus for large systems containing multiple harmonic sources, a separate filter may be required for every major harmonic source [3].

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1043-1052, 2004.

An effective way for harmonic elimination is the harmonic compensation by using active power filter. Active power filter detect harmonic current from distorted wave in power line, then generates negative phase current as same as detected harmonic to cancel out the harmonic in power system. Using of the feed forward neural networks is one of the methods for harmonic detection. [6-8].

In this study, the method to apply the Elman's recurrent neural networks [9] for harmonic detection process in active filter is proposed. The feed forward neural networks were also used for comparison. The distorted wave including 5th, 7th, 11th, and 13th harmonics are used to be input signals for these neural networks at the training state. The output layer of network is consisted of 4 units in according to each order of harmonic. By effect of learning representative data, each component of harmonic is detected to each according unit. That means neural network structures can decompose each order of harmonic and detect only harmonic without fundamental wave in the same time.

2 The Method to Detect Harmonic by Using Neural Network

Figure 1 depicts the concept of active power filter. Figure 2 shows the process of the harmonic detection in the active power filter.

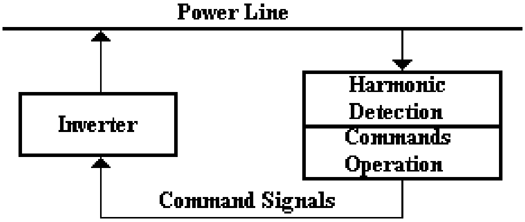


Fig. 1. Concept of active power filter

One method to detect harmonics uses frequency analysis and synthesis. Generally, the distorted current from power line is analyzed in the frequency spectrum. After the component of fundamental wave is eliminated, the remaining harmonics are output to be used for compensating current generation [6-9]. As mentioned before, one of methods to detect harmonic is method by means of feed forward neural networks (FFNN) [6-9]. In this study, we use the feed forward and propose the recurrent neural networks to detect each component of harmonic same as the general method. When the distorted current is detected from power line, the amplitude from one cycle distorted wave is input to each unit of the neural networks in term of serial signals. By means of using the feed forward and recurrent neural networks, each harmonic is decomposed separately without Fourier transformation. At the same time, fundamental wave is eliminated without a low pass filter.

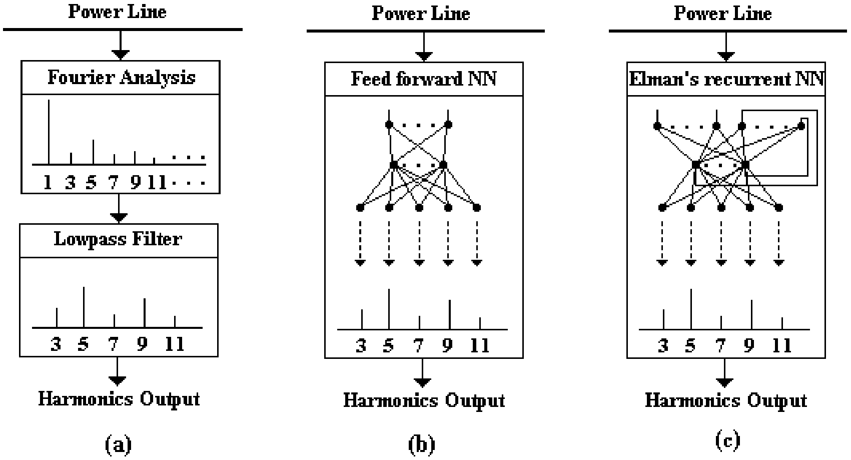


Fig. 2. Process of harmonic detection in active power filter: Fourier analysis type (a), feed forward neural network type (b), Elman's recurrent neural network type (c)

3 Feed Forward Neural Network for Harmonic Detection

Because of non-sinusoidal load currents consisting primarily of lower-order 5th, 7th, 11th, and 13th harmonics that distort the system power quality, we consider about 5th, 7th, 11th, and 13th harmonics detection. At the first step we used the feed forward neural network as seen in figure 3. This network is a multilayer network (input layer, hidden layer, and output layer). The hidden layer neurons and the output layer neurons use nonlinear sigmoid activation functions. In an alternative network, the output layer neurons use linear activation functions for comparison. Equations which used in the neural network model are shown in (1), (2), and (3).

Outputs of hidden layer neurons are,

$$X_j(n) = 1 \bigg/ \left(1 + \exp\left(b_j^h(n) + \sum_{i=1}^{N} W_{ij}^{ik}(n) U_i(n)\right)\right) \tag{1}$$

For sigmoid activation function outputs are,

$$Y_l(n) = 1 \bigg/ \left(1 + \exp\left(b_l^o(n) + \sum_{j=1}^{N1} W_{jl}^{ho}(n) X_j(n)\right)\right) \tag{2}$$

For linear activation function outputs are,

$$Y_l(n) = b_l^o(n) + \sum_{j=1}^{N1} W_{jl}^{ho}(n) X_j(n) \tag{3}$$

where j = 1to $N1$and $N1$is the number of hidden layer nodes, l = 1to $N2$ and $N2$ is the number of output layer nodes, $b_j^h(n)$ are the biases of the hidden layer neurons, $b_l^o(n)$ are the biases of the output layer neurons, $W_{ij}^{ih}(n)$ are the weights from input to hidden layer, $W_{jl}^{ho}(n)$ are the weights from hidden layer to output layer, $U_i(n)$, i = 1 to N are the distorted wave inputs, and $Y_l(n)$, l = 1 to $N2$ are outputs for harmonic coefficients. In this study, N is 128, $N1$ is 90, and $N2$ is 4.

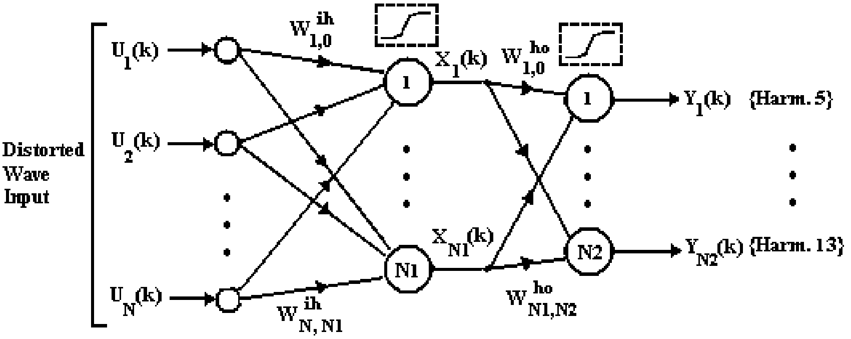


Fig. 3. Feed forward neural network structures for harmonics detection

4 Elman's Recurrent Neural Network for Harmonic Detection

At the second step, because of the time series nature of the distorted wave, we proposed Elman's recurrent neural network [10] for harmonic detection as seen in figure 4. This network is also a multilayer network (input layer, recurrent hidden layer, and output layer). The hidden layer neurons and the output layer neurons use nonlinear sigmoid activation functions. In an alternative network, the output layer neurons use linear activation functions for comparison. Equations which used in the neural network model are shown in (4), (5), and (6).

Outputs of hidden layer neurons are,

$$X_j(n) = 1 \Big/ \left(1 + \exp\left(b_j^h(n) + \sum_{i=1}^{N} W_{ij}^{ik}(n)U_i(n) + \sum_{i=N+1}^{N+N1} W_{ij}^{ik}(n)X_i(n-1)\right)\right) \tag{4}$$

For sigmoid activation function outputs are,

$$Y_l(n) = 1 \Big/ \left(1 + \exp\left(b_l^o(n) + \sum_{j=1}^{N1} W_{jl}^{ho}(n)X_j(n)\right)\right) \tag{5}$$

For linear activation function outputs are,

$$Y_l(n) = b_l^o(n) + \sum_{j=1}^{N1} W_{jl}^{ho}(n) X_j(n) \tag{6}$$

where j = 1to $N1$and $N1$is the number of hidden layer nodes, l = 1to $N2$ and $N2$ is the number of output layer nodes, $b_j^h(n)$ are the biases of the hidden layer neurons, $b_l^o(n)$ are the biases of the output layer neurons, $W_{ij}^{ih}(n)$ are the weights from input to hidden layer, $W_{jl}^{ho}(n)$ are the weights from hidden layer to output layer, $U_j(n)$, i = 1 to N are the distorted wave inputs, X_j $(n-1)$ i = 1 to $N1$are the time delayed outputs of the hidden layer nodes which are measured at a previous time step $(n-1)$, and Y_l (n), l = 1 to $N2$ are outputs for harmonic coefficients. In this study, N is 128, $N1$ is 90, and $N2$ is 4.

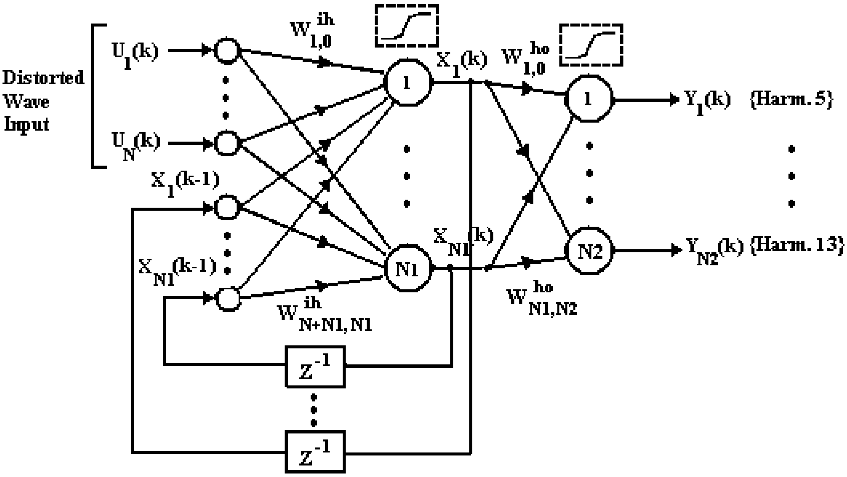


Fig. 4. Elman's recurrent neural network structures for harmonics detection

5 Training of the Networks

A back propagation (BP) method is widely used as a teaching method for an ANN. The main advantage of the BP method is that the teaching performance is highly improved by the introduction of a hidden layer [10]. In this paper, BP learning rules with momentum and adaptive learning rate are used to adjust the weights and biases of networks to minimize the sum-squared error of the network. This is done by continually changing the value of the network weights and biases in the direction of steepest descent with respect to the error. The BP with momentum method decreases

BP's sensitivity to small details in the error surface. This helps the training process to avoid being stuck in shallow minima.

Training time can also be decreased by the use of an adaptive learning rate, which attempts to keep the learning rate step size as large as possible while keeping learning stable [11]. These two techniques can be used with BP to make it a faster, more powerful, and more useful learning paradigm [11]. Standard BP algorithm is also used for comparison.

In order to make neural network enable to detect harmonics from distorted wave, it is necessary to use some representative distorted waves for learning. These distorted waves are made by mixing the component of the 5th, 7th, 11th, and 13th harmonics in fundamental wave. For this purpose, 5th harmonic up to 70%, 7th harmonic up to 40%, 11th harmonic up to 10% and 13th harmonic up to 5% were used and approximately 2500 representative distorted waves were generated for training process.

During the training process, the distorted waves were used for recognition. As the result of recognition, output signal from each output unit means the coefficient of each harmonic which is including in the input distorted wave and these harmonics are eliminated from the distorted wave. Equations which used in the elimination process are shown in (7), and (8).

$$V_f(t) = V_d(t) - \sum_h V_h(t) \tag{7}$$

$$V_h(t) = A_h Sin(2\pi ft + \theta) \tag{8}$$

where, $V_f(t)$ is active filtered wave, $V_d(t)$ is distorted wave, h = 5,7,11,13, A_h are coefficients of lower-order 5th, 7th, 11th, and 13th harmonics, f = 50 Hz, θ is phase angle and equal to zero in this study.

6 The Quality of Power System Waves

The common index used to determine the quality of power system currents and voltages are total harmonic distortion (*THD*) [1,2], which is defined as

$$THD = \sqrt{\frac{\sum_2^{\infty} V_h^2}{V_1^2}} \tag{9}$$

where V_h represents the individual harmonics and V_1 is the fundamental component of load wave.

7 Results and Conclusions

The non sinusoidal load currents consist also that the higher order harmonics such as 17th, 19th, etc., but they do not carry any significant current [1]. So, for the performance evaluation of the neural network structures, 5th harmonic up to 70%, 7th harmonic up to 40%, 11th harmonic up to 10% and 13th harmonic up to 5%, 17th harmonic up to 5%, 19th harmonic up to 2.5%, 23rd harmonic up to 2.5%, 25th harmonic up to 2% were used [12] and approximately 250 representative distorted waves were generated as a test set.

For the training and test processes, input signals of the neural networks are the amplitudes of one period of distorted wave. The amplitudes are taken 128 point at regular interval of time axis. The amplitudes are used to be input signals of the neural networks without any pre processing. At the training phase, the higher order harmonics such as 17th, 19th, etc., are ignored for *THD* calculations.

Figure 5 shows the training results of the feed forward neural networks. As seen in this figure, the results of adaptive back propagation algorithm is better then that of standard back propagation algorithm and the neural network whose output layer neurons use nonlinear sigmoid activation functions shows better results.

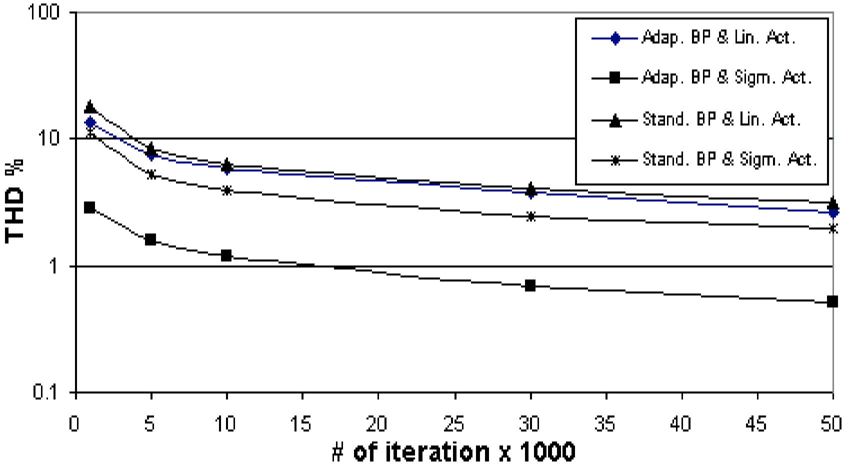


Fig. 5. Training results of feed forward neural networks

Figure 6 shows the training results of Elman's recurrent neural networks. As seen in this figure, the results of adaptive back propagation algorithm is better then that of standard back propagation algorithm and the neural network whose output layer neurons use nonlinear sigmoid activation functions shows better results.

After the training process is completed, the general distorted waves (test set) were used for recognition. As the result of recognition, output signal from each output unit means the coefficient of each harmonic which is including in the input distorted wave and these harmonics are eliminated from the distorted wave.

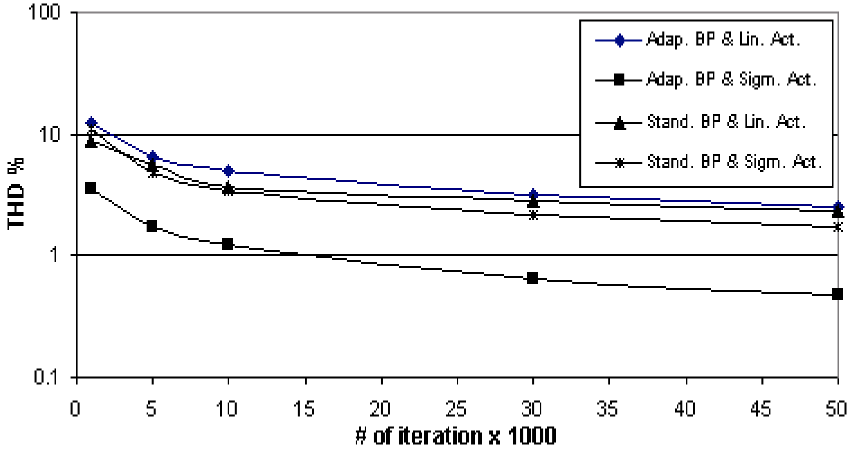


Fig. 6. Training results of Elman's recurrent neural networks

Table 1 shows the average *THD* values of restored waves obtained by using the feed forward and Elman's recurrent neural networks for the test set. The sample source wave and the restored waves are shown in figure 7 and 8.

Table 1. Average *THD* values

Neural Network	Training Algorithm	Activation functions for output neurons	Average *THD* (%)
	Before compensation		46.36
Feed forward NN	Adaptive BP	Linear	4.52
		Sigmoid	3.68
	Standard BP	Linear	4.86
		Sigmoid	4.14
Elman's RNN	Adaptive BP	Linear	4.39
		Sigmoid	3.68
	Standard BP	Linear	4.73
		Sigmoid	4.02

The recommendation IEEE 519 allows a total harmonic distortion (*THD*) of 5% in low-voltage grids [13]. As seen in the table 1, average *THD* value is 46.36% before compensation and obtained average *THD* values are less then 5% after compensation for all networks. These *THD* values are suitable to the recommendation IEEE 519. 3.65% of these *THD* values come from the higher order harmonics such as 17th, 19th, etc which are not used in the training. This means that there is a improvement potential. The *THD* values obtained by using Elman's recurrent neural networks are better than the *THD* values obtained by using the feed forward neural networks. This can be because of that the feedback structures of the Elman's RNN is more appropriate for the time series nature of the waves.

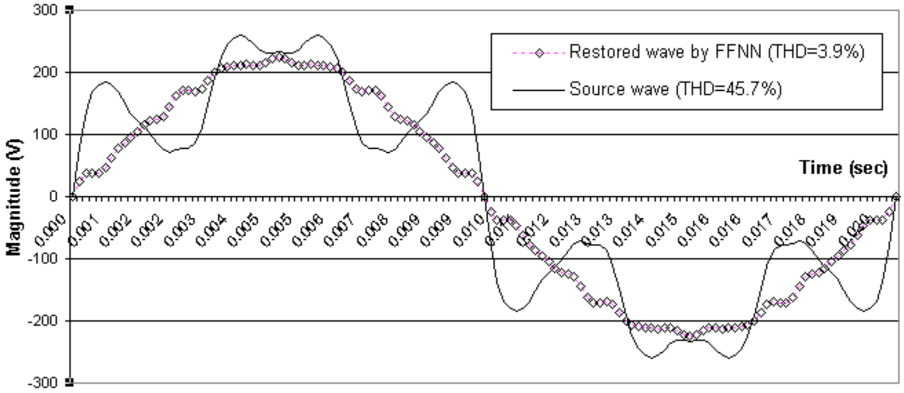


Fig. 7. Sample source and restored waves (by FFNN)

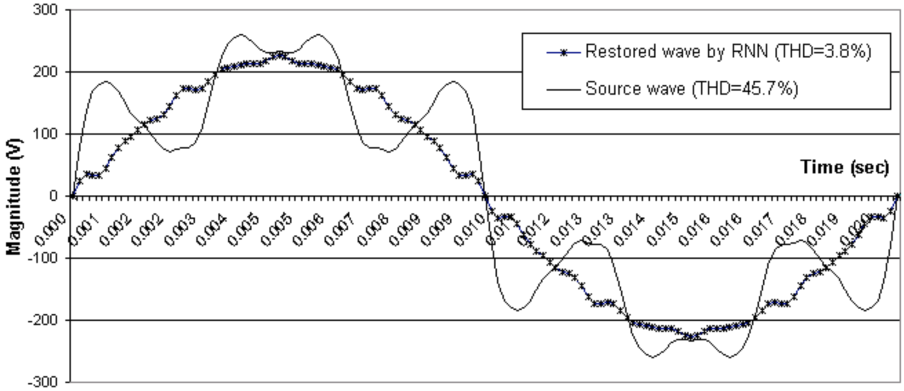


Fig. 8. Sample source and restored waves (by Elman’s RCNN)

As the result, the possibility of the feed forward and Elman’s recurrent neural networks to detect harmonics is confirmed by compensating the distorted waves and it can be said that the feed forward and Elman’s recurrent neural networks are effective to use for active filter.

References

1. Ryckaert, W.R.A., Ghijselen, J.A.L., Melkebeek, J.A.A.: Harmonic mitigation potential of shunt harmonic impedances, Electric Power Systems Research, Vol. 65 (2003) 63-69
2. Rastegar, S.M.R., Jewell, W.T.: A new approach for suppressing harmonic disturbances in distribution system based on regression analysis, Electric Power Systems Research Vol. 59 (2001) 165-184

3. Unsal, A., Von Jouanne, A.R., Stonic, V.L.: A DSP controlled resonant active filter for power conditioning in three phase industrial power system, Signal Processing, Vol. 82 (2001) 1743-1752
4. IEEE Standarts 519-1992, IEEE Recommended Practice and Requirements for Harmonics Control in Electric Power Systems, Piscataway, NJ, (1992)
5. IEEE Recommended Practices for Power System Analysis, IEEE Inc., New York, NY (1992)
6. Pecharanin, N., Sone, M., Mitsui, H.: An application of neural network for harmonic detection in active filter, ICNN (1994) 3756-3760
7. Rukonuzzaman, M., Nakaoka, M.: Adaptive neural network based harmonic detection for active power filter, IEICE Transactions On Communications, E86B (5) (2003) 1721-1725
8. Gunturkun, R., Yumusak, N., Temurtas, F.: Detection of Harmonics by using Artificial Neural Network, The IJCI Proceedings, Vol. 1(1), TAINN'03, July (2003)
9. Gunturkun, R., Temurtas, F., Yumusak, N., Unsal, A., Temurtas, H.: Compensation of Harmonics by using Artificial Neural Networks, 3rd Int. Adv. Tech. Symp., August (2003)
10. Abdelhameed, M.M. Tolbah, F.F.: A recurrent neural network based sequential controller for manufacturing automated systems, Vol. 12, (2002) 617-633
11. Haykin, S.: Neural Networks, A Comprehensive Foundation, Macmillan Publishing Company, Englewood Cliffs, N.J. (1994)
12. Reid, W.E.: Power quality issues – standards and guidelines, IEEE Trans. on Ind. App., Vol. 32(3) (1996) 625- 632
13. Nunez-Zuniga, T.E., Pomilio, J.A.: Shunt active power filter synthesizing resistive loads, Vol. 17(2) (2002) 273-278

Design of an Adaptive Artificial Neural Network for Online Voltage Stability Assessment

Mehran Rashidi[1] and Farzan Rashidi[2]

[1] Hormozgan Regional Electric Co. Bandar-Abbas, Iran,
P.O.Box: 79167-95599, Bandar-Abbas
mrashidi@mehr.sharif.edu
[2] Control Research Department, Engineering Research Institute, Tehran, Iran,
P.O.Box: 13445-754, Tehran
f.rashidi@ece.ut.ac.ir

Abstract. Voltage instability has become a major concern in many power systems and blackouts have been reported, where the reason has been voltage instability. Voltage stability is concerned with the ability of a power system to maintain acceptable voltages at all buses in the system under normal condition and after being subjected to a disturbance. It is an important consideration in the design and operation of power system. Voltage stability is also called the load stability. It is a characteristic of a power system, which is required to transmit sufficient power to meet load demand. The objective of this paper is to present the application of artificial neural network (ANN) in on-line assessment of voltage stability. The proposed method is radial basis function (RBF) neural network. This NN is used for estimation of voltage stability margins (VSM). The IEEE-118 test system is considered for application to this method. A comparison between the proposed NN and a multi-layer perceptron (MLP) with standard error back-propagation learning (EBPL) is presented, which indicates efficiency of this NN. Obtained results confirm the validity of the proposed approach.

1 Introduction

Voltage stability is concerned with the ability of a power system to maintain acceptable voltages at all buses in the system under normal condition and after being subjected to a disturbance [1]. It is an important consideration in the design and operation of power system. Voltage stability is also called the load stability. It is a characteristic of a power system, which is required to transmit sufficient power to meet load demand. A power transmission network has an inherent limit as to how much power it can deliver to loads. When this limit is exceeded, the voltages experienced by loads become too low to be practically useful. In many cases, the voltages will go straight from normal to zero in a matter of a few seconds or minutes. This process is called voltage collapse.

Voltage instability is a problem in power systems which are heavily loaded, faulted or have a shortage of reactive power. The nature of voltage instability can be analyzed by examining the production, transmission, and consumption of reactive power. The

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1053-1061, 2004.

problem of voltage instability concerns the whole power system, although it usually has a large involvement in one critical area of the power system. Power system is voltage stable if voltages after a disturbance are close to voltages at normal operating condition. According to [2] the definition of voltage instability stems from the attempt of load dynamics to restore power consumption beyond the capability of the combined transmission and generation system. The voltage stability may be divided into short and long term voltage stability according to the time scale of load component dynamics. Short term voltage stability is characterized by components such as induction motors, excitation of synchronous generators, and electronically controlled devices such as HVDC and SVC [2]. When short term dynamics have died out some time after the disturbance, the system enters a slower time frame. The long term voltage stability is characterized by scenarios such as load recovery by the action of on load tap changer or through load self restoration, delayed corrective control actions such as shunt compensation switching or load shading. The long term dynamic such as response of power plant controls, boiler dynamics and automatic generation control also affect long term voltage stability. In some cases the monitoring of power system security becomes more complicated because the critical voltage might be close to voltages of normal operation range. Therefore, a tool which can provide timely evaluation of voltage stability of the system under the diversified operating conditions would be very useful.

For purposes of analysis, it is sometimes useful to classify voltage stability into small and large disturbances. Small disturbance voltage stability considers the power system's ability to control voltages after small disturbances, e.g. in load [3]. The analysis of small disturbance voltage stability is done in steady state. In that case the power system can be linearalized around an operating point and the analysis is typically based on eigenvalue and eigenvector techniques. Large disturbance voltage stability analyses the response of power system to large disturbances e.g. faults, switching or loss of load, or loss of generation. Large disturbance voltage stability can be studied by using non linear time domain simulation in the short time frame and load flow analysis in long term time frame [4]. The voltage stability is, however, a single problem on which a combination of both linear and non linear tools can be used. References [2, 3, 5, 6] provide a good overview of these areas. There has developed a wide variety of modeling principles and of computation and control methods to analyse and control power system voltage stability. The research has mainly based on analytical methods such as dynamic simulations and load flow. The assessment of voltage stability is commonly based on voltage stability margin, but the computation of the margin is time consuming. The computation methods developed for voltage stability analysis are best suited for power system planning. Online voltage stability assessment has mainly also been based on these computation method. The application of pattern recognition, decision tree, fuzzy systems, genetic algorithms and statistical methods are also applied to on line voltage stability assessment. Instead of laws of physics and logic these methods create or use a model, which is based on knowledge or data. Statistical method is black box model, which is applied when the functioning of a system is unknown or very complex, but there is plenty of data available. The black boxes are data driven models, which do not explicitly model the physics of system, but establish a mathematical relationship between a large numbers of input-output pairs measured from the system. The mathematical relationship is a model, which is numerical computed from measurement or statistical data. Nero fuzzy and expert systems, which are gray box models, may be applied when there is also some knowledge available. Many good voltage stability indices and methods have been created for power systems. The most common methods to estimate the proximity of voltage collapse point are minimum singular value [7], point of collapse method [8, 9] optimization [10, 11] and continuation load flow [12, 9, 13]. Other voltage stability indices are also proposed in the literature like sensitivity analysis [14], second order performance index [15], the energy function method [16], modal analysis [17] and many others which can be found in references [6, 18]. The energy method has been used for voltage security assessment [19, 20]. The energy margin, as an indicator of system voltage security is physically understandable. A clear conclusion of the description of the above computation methods is that all the methods have different function. They have their own strengths and weaknesses. The combination of all of these methods gives most information in voltage stability analysis. The minimum singular value, the continuation load flow and the maximum loading point methods compute intermediate results of the PV curve. These methods provide voltage stability indices for quantifying proximity to the voltage collapse point and identifying the weakest buses. The point of collapse method and the optimization method compute the exact voltage collapse point using a direct method. These methods also provide information related to voltage weak areas and to voltage stability enhancement.

This paper proposes an artificial neural network (ANN) approach in on-line voltage stability assessment. Details of this NN are explained and its efficiency is demonstrated. The energy method for voltage security assessment was adopted in this paper as the benchmark method to generate sample patterns for training and testing of the designed NN. In NN applications, selecting of input variables is an important aspect. In this paper, input variables are determined in a systematic way in terms of their discriminating capability.

2 The Proposed Neural Network

The radial basis function (RBF) network is the feedforward neural network scheme that uses local type of activation function. In the simplest form, a radial basis function network consists of three layers; the input layer, which has source (input) nodes, the hidden layer, which has enough number of neurons, and the output layer, which defines the response of the network with regard to the applied inputs. The mapping from the input layer to the hidden layer is nonlinear, whereas the mapping from the hidden layer to the output layer is linear. The structure of the RBF network is shown in Fig 1. In this network, the radial basis functions are Green functions. Most often used type is the gaussian function defined in the way:

$$\varphi(\mathbf{x}) = \exp\left(-\frac{\left\| \mathbf{x} - \mathbf{c} \right\|^2}{\sigma^2} \right) \tag{1}$$

in which **c** is the center and the parameter **σ** represents the width of the function. This is the local function of the nonzero activation only at the neighbourhood of the center **c**. In contrast to the MLP the RBF networks form the local approximation approach. The main advantage of it is very clear physical association of the parameters of the radial neurons and the input data. In RBF networks the representation of particular data point is done by very limited number of neurons, specializing in this task. The other neurons do not contribute to this particular data and are associated with the other data in different region of the space. The output neuron of the network is usually linear and performs simple weighted summation of the output signals of all hidden neurons, i.e.:

$$y = w_0 + \sum_{i=1}^{K} w_i \varphi_i \tag{2}$$

The learning method of the RBF network belongs to the supervised type, in which the input vector **x** and the destination *d* form the learning pairs ($\mathbf{x}_i$, d_i) for i= 1, 2, ..., p. Usually the learning task of RBF network is split into two separate subtasks. The first is determination of the radial function parameters and the second - the calculation of the output weights w_i. Many different strategies may be applied for center adjustment. For example some approaches assume at the beginning as many centers as is the number of data ($\mathbf{c}_i$=$\mathbf{x}_i$) and then in the process of learning they reduce this number sequentially according to their contribution to the final representation of the data. In this paper we adopt “Winner Takes All” (WTA) strategy. In this algorithm the centers are associated with the clusters of input data, represented by the average vector $\mathbf{x}_i$. The self-organization of this clustering process is done by the competition among neurons. The self-adaptation process in this approach is performed only for the winner, that is the neuron, whose weight vector $\mathbf{c}_i$ is closest to the input vector $\mathbf{x}_k$, and can be described by the relation

$$\mathbf{c}_i(k+1) = \mathbf{c}_i(k) + \eta[\mathbf{x}_k - \mathbf{c}_i(k)] \tag{3}$$

of η the learning coefficient changing in time from η_{max} to zero. After adaptation of the centers the adjustment of the width parameter σ_i is done in a way to provide the continuous approximation of the data. Most often used method is the so-called P-neighbours approach, according to which σ_i is proportional to the averaged distances of $\mathbf{c}_i$ to its P closest neighbours

$$\sigma_i = \sqrt{\frac{1}{P} \sum_{k=1}^{P} \left\| \mathbf{c}_i - \mathbf{c}_k \right\|^2} \tag{4}$$

After determination of the parameters of the radial functions the weights of the output neurons are calculated in the next stage. The easiest way is to apply of the singular value decomposition (SVD) to the matrix describing the signals in the network. Observe that the output signals of the RBF network for all p training pairs may be arranged in the matrix form as follows

$$\mathbf{FW=d} \tag{5}$$

where

$$\mathbf{d}=\left[d_1,d_2,..,d_p\right]^T \quad \mathbf{w}=\left[w_0,w_1,..,w_K\right]^T \quad \mathbf{F}=\begin{bmatrix} 1 & \varphi_1(\mathbf{x}_1) & ... & \varphi_K(\mathbf{x}_1) \\ 1 & \varphi_1(\mathbf{x}_2) & ... & \varphi_K(\mathbf{x}_2) \\ ... & ... & ... & ... \\ 1 & \varphi_1(\mathbf{x}_p) & ... & \varphi_K(\mathbf{x}_p) \end{bmatrix} \tag{6}$$

This is the rectangular system of equations that can be solved using the pseudo-inverse $\mathbf{F}^+$ of the matrix **F**

$$\mathbf{w}=\mathbf{F}^+\mathbf{d} \tag{7}$$

The pseudoinverse is usually calculated using SVD technique. The process of training the RBF network is usually done in the iterative way. After adjusting the parameters of the radial functions using the self-organization, the weight vector **w** is calculated using (7) and then the error function **e=y-d** determined. This error is back propagated to the hidden layer and the positions of centres and widths of the radial functions are corrected according to the gradient methods, in a similar way as it was done in MLP networks. This process is repeated many times until the stabilization of all parameters of the network.

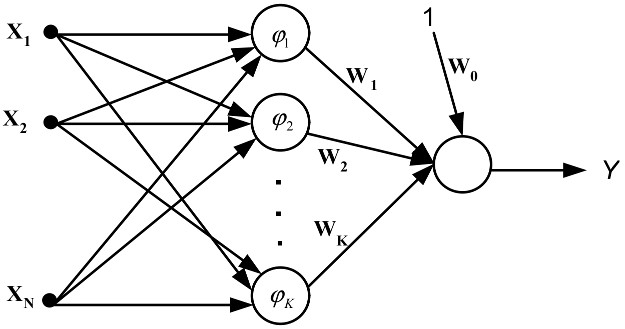


Fig. 1. The general structure of the RBF neural network

For introducing the learning phase, in addition to the learning algorithm, training samples, i.e. input and output variables of the NN must be determined. Appropriate selection of input variables is the key to the success of NN applications. Usually heuristic knowledge is required in choosing input variables. Voltage stability of a power system is mainly affected by the system's loading conditions. Using the energy method, loading conditions are utilized to find the corresponding stable equilibrium point (SEP) and unstable equilibrium points (UEP) by solving the load flow equations. The procedure can be expressed as a mapping:

$$\left(P,Q,V_g\right)\rightarrow\left(V^s,\alpha^s,V^u,\alpha^u\right) \tag{8}$$

Where *P, Q* and V_g are specified bus net real and reactive power injections and generator bus voltage magnitudes vectors, respectively. $x^s=\left(V^s,\alpha^s\right)$ is the vector of the SEP and $x^u=\left(V^u,\alpha^u\right)$ is the vector of the UEP. Energy margins are computed by substituting x^s and x^u in the energy function. System voltage stability is described by the minimum value of all the obtained margins. This process can also be described by another mapping:

$$\left(V^s, \alpha^s, V^u, \alpha^u\right) \rightarrow \text{Energy Margins} \tag{9}$$

The above mapping shows that variables *P, Q* and V_g govern the voltage stability of a power system. They can be chosen as inputs to the NN. For the VSM estimation problem, one output is required. Finally, the way of presentation of training samples to the NN must be determined. In [21], two modes of NN training for voltage stability assessment have been studied, sequential and random presentation of training patterns. The sequential presentation calls for generating pattern and they are presented to the NN for training in that same order. In random presentation a pre-specified number of training patterns are generated randomly within the specified loading ranges and are presented to the NN in a random order. Equivalent NN performances for the two training modes were obtained. Due to its flexibility, random generation of training samples is adopted in this paper. After training information recalling by the NN is performed through feed-forward processing, by using (2), of input information of test patterns.

3 The Energy Method for Voltage Stability Assessment

The energy method uses an energy function, derived from a closed from vector integration of the real and reactive power mismatch equations between a SEP and an UEP, to provide a quantitative measure of system vulnerability to voltage instability. The energy function for voltage stability assessment, as derived in [20], is expressed as below:

$$\begin{aligned} SM(x^S, x^u) = & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[B_{ij}V_i^u V_j^u \cos\alpha_{ij}^u - B_{ij}V_i^s V_j^s \cos\alpha_{ij}^s\right] \\ & -\sum_{i=1}^{n}\left[\int_{V_i^s}^{V_i^u}\frac{Q_i(x)}{x}dx + P_i(\alpha_i^u - \alpha_i^s)\right] \\ & -\sum_{i=1}^{n}[\left(\alpha_i^u - \alpha_i^s\right)\sum_{j=1}^{n} G_{ij}V_i^s V_j^s \cos\alpha_{ij}^s \\ & \qquad + \frac{V_i^u - V_i^s}{V_i^s}\sum_{j=1}^{n} G_{ij}V_i^s V_j^s \sin\alpha_{ij}^s] \end{aligned} \tag{10}$$

Where V_i and α_i are voltage magnitude and phase angle at bus *i*, P_i and Q_i are net real and reactive power injections at bus *i*, respectively (reactive power can be voltage-dependent); G_{ij} and B_{ij} are the real and imaginary parts of the *ij*-th element of the bus admittance matrix; *n* is the number of system buses. The UEPs of the system in (10), which corresponds to a SEP are governed by the power flow equations. There are generally a total of 2^{n-1} UEPs [22]. This reduces the number of UEPs to n-1 UEPs must be found out in order to evaluate system voltage security. Computation of all the n-1 UEPs can be a challenging task for large systems. Mapping from system loading conditions to SEPs and UEPs, and from SEPs and UEPs to voltage stability margins are constructed in (8) and (9). Further, it is assumed that there is a direct mapping relation between the operating states and the VSMs, i.e.:

$$\left(P,Q,V_g\right) \rightarrow \text{Minimum Energy Margins} \tag{11}$$

An NN is then used to exploit the mappings. VSMs for specific operating state can be calculated directly from the trained NN.

4 Numerical Results

This section includes applications of the proposed approach for IEEE 118 bus test system. The system data of the IEEE test systems can be found in [28]. In the initial studies, real and reactive powers at all buses were applied as inputs to the NN. In this case, a relatively large NN was built. Also the NN training was time-consuming. For practical size systems, this scheme is not feasible due to the large number of loads and generations. In this study, input information to the NN is refined using the sensitivity analysis. It was noted that real and reactive powers with larger sensitivities have more significant influence on voltage stability. This fact also implies that these variables have stronger discriminating capabilities for voltage stability evaluation. As a result, bus real and reactive powers with larger sensitivities are selected as NN inputs. Extensive simulation studies verified that this scheme performs quite satisfactory. Using these variables as NN inputs, the mapping from loading conditions to VSMs can be set up, even for large systems, with a reasonable number of inputs. The input variables identified based on sensitivity analysis for the test system are listed in Table 1, where P_{gi} and Q_{gi} refer to real and reactive power generations at bus P_{di} and Q_{di} are real and reactive powers of the load at bus j respectively. The choice of reactive power outputs of generators as NN inputs is more appropriate than voltage. And Q_gs instead of V_gs are listed in Table 1. The most vulnerable bus of the IEEE 118 bus system is 34.

Table 1. NN input variables selected based on sensitivity for the test system

Number of inputs	Input Variables
32	Pd34, Pd76, Pd82, Pd83, Pd84, Pd86, Pd88, Pd93, Pd94, Pd95, Pd102, Pd106, Pd109, Pd112, Qd4, Qd40, Qd49, Qd67, Qd74, Qd76, Qd80, Qd108, Qg19, Qg32, Qg42, Qg44, Qg46, Qg76, Qg104, Qg105, Qg111, Qg113

A number of test samples, which all of them are different from the training patterns, are created for the test system according to the described procedures for the training patterns. MLP with the standard EBPL is one of the most widely used NNs in the power system applications such as voltage stability assessment [17-22]. Obtained results from this NN and the proposed NN, for the test system, are shown in Table 2 (both NNs have the same training and test samples). These results indicate efficiency of the designed NN, which with less training times has much more estimation accuracy for VSM.

Table 2. Test results, Comparison of proposed method and MLP neural network

Training time (designed NN)	Test error (designed NN)	Training time (MLP with EBPL)	Test error (MLP with EBPL
697 sec	0.37%	1274	5.54%

5 Conclusion

Voltage stability problems, once associated primarily with weak system and long lines, are currently a source of concern in highly developed systems as a result of heavy loading. Operators must be able to recognize voltage stability related symptoms, so that appropriate remedial action can be taken. Therefore, on-line monitoring and analysis to identify potential voltage stability problems would be invaluable in this regard. In the work reported upon in this paper, the emphasis has been on investigation whether the complex relationship between power systems operating states and the corresponding stability margins can be captured, for on-line security monitoring purpose, by the NN technique. For this purpose, a new NN with high estimating power has been designed. This NN is a RBF neural network with WTA strategy. Comparison of this NN with a MLP with the EBPL, which has been used in the previous works, indicates efficiency of the proposed NN. In this work, the mapping from power system loading conditions to VSMs is constructed using the energy method. Determination of input variables has been a key issue in NN application. A NN using thus selected variables as inputs possesses smaller topology with sufficient discriminating capability. This feature enables NN training to be done within a reasonable time. More importantly it facilitates the NN application for large systems. Extensive testing on the IEEE confirms the validity of the developed approach.

Reference

[1] P. Kundur, Power system stability and control, NewYork: McGraw-Hill 1993.

[2] Van Cutsem T. and Vournas C., Voltage stability of electric power systems, Kluwer academic publishers, Boston, USA, 1998

[3] X. Ma, A. A. El-Kieb, R. E. Smith and H. Ma, Estimation of Voltage Stability Margins Using Artificial Neural Networks, Proc of the 26th NAPS, Sept. 1994

[4] Gao B. et al., Towards the development of a systematic approach for voltage stability assessment of large scale power systems, IEEE Trans. On Power Systems, Vol. 11, No. 3, August 1996, pp. 1314-1324

[5] Taylor C.W., Power system voltage stability, McGraw-Hill, New York, USA, 1994

[6] IEEE/PES Power System Stability Subcommittee Special Publication, Voltage stability assessment, procedures and guides, Final draft available at http://www.power.uwaterloo.ca, January 2001

[7] Lof P-A., On static analysis of long term voltage stability, PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1995.

[8] Cnizares C.A et al., Point of collapse method applied to AC/DC power systems, IEEE Trans. On Power Systems, Vol. 7, No. 2 May 1992, pp.673-680.

[9] Canizares C.A and Alvarado F.L., Point of collapse and continuation methods for large AC/DC systems, IEEE Trans. On Power Systems, Vol. 8, No. 1 February 1993, pp.1-8

[10] Obadina O.O and Berg G.J., Determination of voltage stability limit in multi machine power systems, IEEE Trans. On Power Systems, Vo. 3, No. 4, November 1988, pp. 1545-1552

[11] Van Cutsem T, A method to compute reactive power margins with respect to voltage collapse, IEEE Trans. On Power Systems, Vo. 6, No. 1, February 1991, pp. 145-153

[12] Ajjarapu V. and Christ C., The continuation power flow: a tool for steady state voltage stability analysis, IEEE Trans. On Power Systems, Vo. 7, No. 1, February 1992, pp. 416-423

[13] Canizares C.A. and Alvarado F.L., UWPFLOW continuation and direct methods to locate fold bigurcation in AC/HVDC/FACTS power system, the software is available at http://iliniza.uwaterloo.ca

[14] Flatabo N. et al., A method for calculation of margins to voltage instability applied on the Norwegian system for maintaining required security level, IEEE Trans. On Power Systems, Vo. 8, No. 3, August 1993, pp. 920-928.

[15] Berizzi A. et al., First and second order methods for voltage collapse assessment and security enhancement, IEEE Trans. On Power Systems, Vo. 13, No. 2, May 1998, pp. 543-549.

[16] Overbye T.J., Use of energy methods for online assessment of power system voltage security, IEEE Trans. On Power Systems, Vo. 8, No. 2, May 1993, pp. 452-458

[17] Gao B. et al., Voltage stability evaluation using modal analysis, IEEE Trans. On Power Systems, Vo. 7, No. 4, November 1992, pp. 1529-1536

[18] Ajjarapu V. and Lee B., Bibliogarphy on voltage stability, IEEE Trans. On Power Systems, Vo. 13, No. 1, February 1998, pp. 115-125

[19] C. L. DeMarco and T. J. Overbye, An Energy Based Security Measure for Assessing Vulnerability to Voltage Collapse, IEEE Trans. On Power Syst., Vol.5, No.2, May 1990

[20] T. J. Overbye, Application of an Energy Based Security Method to Voltage Instability in Electrical Power Systems, Dissertation, University of Wisconsin-Madison, 1991

[21] X. Ma, A. A. El-Kieb, R. E. Smith and H. Ma, Estimation of Voltage Stability Margins Using Artificial Neural Networks, Proc of the 26th NAPS, Sept. 1994

[22] I. Dobson and H. Chaing, Towards a Theory of Voltage Collapse in Electric Power Systems, Systems & Control Letters, 1989

Mining Multivariate Associations within GIS Environments

Ickjai Lee

School of Information Technology,
James Cook University, Douglas Campus,
QLD4811, Australia
Ickjai.Lee@jcu.edu.au

Abstract. As geospatial data grows explosively, needs for the incorporation of data mining techniques into Geographic Information Systems (GISs) are in great demand. Association rules mining is a core technique in data mining and is a solid candidate for the cause-effect analysis of large geospatial databases. It efficiently detects frequent asymmetric causal patterns in large databases. In this paper, we investigate a series of geospatial preprocessing steps involving data conversion and classification so that traditional boolean and quantitative association rules mining can be applied. We present a robust geospatial multivariate association rules mining framework for efficient knowledge discovery within data-rich GISs environments. The proposed approach can be integrated into traditional GISs using dynamic link library and scripting languages such as AVENUE for ArcView and MapBasic for MapInfo. Our framework is designed and implemented in AVENUE for ArcView GIS. Experiments with real datasets demonstrate the robustness and efficiency of our approach.

1 Introduction

Understanding human behaviors and our daily life phenomena is a major key to the success of many businesses. Since these human activities and natural phenomena are georeferenced and geospatially distributed, data analysis seeking geospatial patterns is a crucial task. GISs are computer-based information systems particularly designed for the manipulation and analysis of geospatial databases. They model the real world along with many different geographical layers (themes or coverages) [12]. Each layer captures some unique feature in it in the form of point, line or area. In this multi-layered view, GISs are structured as a number of layers containing heterogeneous data types. GISs provide various layer-based local, focal and zonal functions to help users explore and gain insights into complex geospatial databases. In such multi-layered architecture, it is important to find causal relationships among layers for the analysis of thematically associated events. Detected positive causal patterns are key resources for planning, geospatial decision-making, precaution and prediction.

Several geospatial statistics are available within traditional GISs including χ^2*-test*, *nearest neighbor distances*, *the K-function*, *Moran's I* and *Geary's c* statistics [4] to

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1062-1071, 2004.

detect geospatial correlation structure (geospatial dependence or autocorrelation). However, these correlation statistics have some common limitations. First, they are computationally inefficient. That is, they are not applicable to data-rich environments [13]. Second, they are limited to symmetric relationships. Thus, it is impossible to find asymmetric causal factors with these geospatial statistics. As geospatial data grows explosively, the incorporation of efficient data mining techniques into GISs is in high demand. Association rules mining is one of popular techniques in data mining. It efficiently detects frequent asymmetric causal patterns in large databases. Thus, it is a solid candidate for the asymmetric cause-effect analysis of large geospatial databases.

In this paper, we investigate an efficient and effective geospatial knowledge discovery framework. It consists of a series of geospatial preprocessing steps involving data conversion and categorization. Heterogeneous data types are transformed in such a way that traditional association rules mining can be easily applied. The proposed geospatial association rules mining framework considers the characteristics of multi-layered GISs and uncovers unexpected but valuable causal patterns. The framework can be integrated into traditional GISs using dynamic link library and scripting languages such as AVENUE for ArcView GIS and MapBasic for MapInfo GIS.

The rest of paper is organized as follows. Section 2 reviews association rules mining in the light of geospatial context. Section 3 outlines the framework of areal categorized association rules mining for multi-layered GISs environments. It provides details of data conversion and categorization schemes. Section 4 conducts a serious of experiments with real crime datasets and summarizes experimental results. Section 5 concludes with final remarks.

2 Association Rules Mining

Association rules mining has been one of the most popular pattern discovery methods in the data mining community. It is originated from the market basket analysis within transactional databases. It searches for frequent interesting relationships, positive associations and causal patterns among items. One possible pattern could be □customers who purchase mushroom are likely to purchase beef at big supermarket chains□ An association rule is an expression in the form of $X \Rightarrow Y$ (*c*%), where *X* is the *antecedent* and *Y* is the *consequent*, *X* and *Y* are sets of items ($X \cap Y = \phi$) in transactional databases. It is interpreted as □*c*% of data that satisfy *X* also satisfy *Y*□□ An association rule is typically measured by statistical significance and its strength. *Support,* indicating statistical significance, is the probability that *X* and *Y* exist in a transaction *T* in database *D*. *Confidence* showing the rule's strength is the probability that *Y* exists in a transaction *T* given that *T* contains *X*. That is, support is an estimate for *Prob*($X \cap Y$) and confidence is an estimate for *Prob*($Y|X$). A set of items is refereed to as an *itemset*. Two user defined thresholds, *minimum support* and *minimum confidence*, are used for pruning rules to find only interesting rules. Itemsets satisfying the required minimum support are named *frequent* while rules satisfying the two thresholds are called *strong*. Association rules mining is to compute all strong rules satisfying two user-specified minimum support and minimum confidence constraints. Here, *boolean association rules mining* attempts to correlate the presence of a set of items with another set of items ignoring quantitative values. However, *quantitative association rules mining* [18] is to consider quantities. It discretizes numerical values into several categories (classes) and then boolean association rules mining is applied.

Several attempts have been made to apply association rules mining to the geospatial context. Koperski and Han [11] proposed *spatial association rules mining* that is based on concept hierarchies. It adopts a progressive deepening approach that traverses concept hierarchies from top to bottom to gradually prune the search space. Only strong rules at higher conceptual levels can be further mined at lower levels. For instance, a rule with a *very_close_to* predicate can be mined only if the rule above with *close_to* is strong. However, this approach requires extensive experts□prior knowledge and excessive user involvement. The former is required to build various concept hierarchies and the latter is needed to focus on predicates of user□s interest. Thus, this approach can efficiently confirm what users expect, but hardly reveal totally unexpected hidden patterns which is one of main goals in data mining. Recently, Shekhar and Huang [17] developed *co-location rule mining* that discovers a subset of features given a set of point features frequently located together in a geospatial context. This approach extends traditional association rules mining to the set of transactions is a continuum in a space. Note that, due to the continuity of geospatial space, there may not be an explicit finite set of transactions. In this approach, point locations are mapped to transactions while geospatial features are mapped to items. Since features have different occurrences, the size of transactions varies with features. Thus, the traditional interesting measures are no longer applicable. Co-location rules mining introduces two new interesting measures that can be used in a dynamic situation where transactions are not fixed to a constant. However, the applicability of this approach is limited. This approach only considers 0-dimensional boolean point data type. It is not directly applicable to 1-dimensional line and 2-dimensional area data types. More complicatedly, GISs consist of multiple layers that contain heterogeneous data types. It is noted that metric, directional and topological relationships are also more complex with dimensional objects [16]. Co-location rules mining must be able to handle different dimensional data types simultaneously to uncover complex causal relationships.

3 Areal Categorized Association Rules Mining

3.1 Background

In our approach, we mine multivariate associations based on an areal base layer. This is due to several reasons. First, converting complex data types to primitive data types may cause loss of information. For instance, converting an areal object to a point object looses some information such as area, perimeter or shape. Also, point data analysis is relatively difficult while areal data analysis is well established [4,5]. In addition, existing GIS functions (intersection, containment and buffering) support transforming primitive data types to complex data types. Second, we are often required to analyze geospatial data on an area basis. In particular, this is the case when area aggregates are only of interest such as in voting data and census data. Also, this is the case when specific site measurements are not available for some reason. In socio-economic and demographic measures, the areas often are the only spatial units at which we can collect the feature values. Third, it is pointed out that data mining may threaten privacy and security [3]. Privacy preserving data mining has gained a lot of research attention [3]. In geospatial crime data, crime incidents are aggregated on an area basis to avoid delicate privacy and security issues. It is noted that area aggregates at quite small geographical scales may remove privacy issues and provide still useful information [4]. Thus, area aggregated data can reveal general patterns while preserving privacy. Fourth, topology is indispensable to GISs. Difficulties of topological operations with points have been pointed out [4,9], but instead topological operations with areal objects have been identified as relatively easy and well established [5].

3.2 Working Principles

Our framework is a three-phase knowledge discovery process. It initially involves two preprocessing processes called data conversion and categorization. After that data mining is applied to the preprocessed datasets. In principle, it transforms all geosptial layers into areal aggregated data types based on an areal base map. Areal aggregates are numerical values that will be further categorized into several groups. Normally, GISs provide a number of categorization schemes. We use built-in categorization functions within GISs in this paper. Fig. 1 depicts the overall process involving data conversion, categorization and association rules mining. It is interactive and iterative. Each will be further discussed in the subsequent sections.

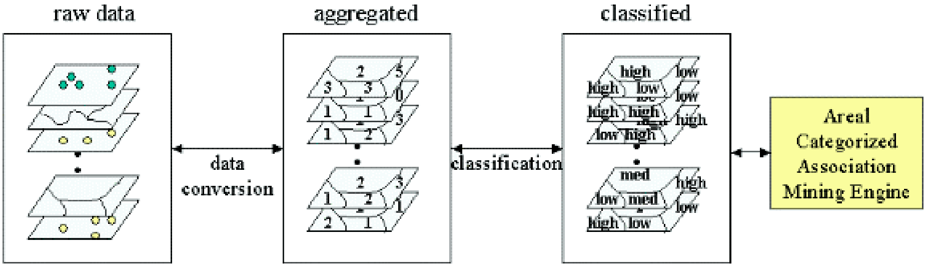


Fig. 1. Areal categorized geospatial association rules mining processes.

3.3 Data Conversion

In geometrical and topological transformations, *intersection* operation is a basis to other geospatial operations such as *meet, overlap*, *cover* and *contain* [19]. Thus, intersection is a fundamental operation and is used as default for conversion here.

3.3.1 Point-to-Area Data Conversion

Point-to-area conversion has been widely used in GISs. *Containment* or *within* function, so called *point-in-polygon* operation, can be used to aggregate point data. Aggregates are typically visualized through choropleth maps using graduated color to represent density. Fig. 2 shows a choropleth map generated by ArcView GIS with 5 classes. In this paper, point-in-polygon operation is used in point data aggregation.

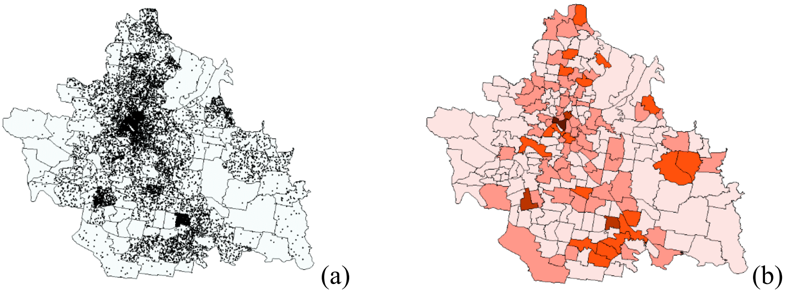


Fig. 2. Point-to-area data conversion: (a) A set of point data with an areal base map; (b) A choropleth map with graduated color representing density.

3.3.2 Line-to-Area Data Conversion

Similar to point-to-area conversion, we can use the intersection topological relationship in line-to-area conversion. Areal units that intersect with a target line object would represent the presence of the line object. Thus, those areal units imply proximity to the line object. This approach would work well when areal units are reasonably largely scaled to capture the details. One alternative is to use buffering. Buffering creates an enclosing polygon of a line object at a specified distance. Once transformed, we can apply area-to-area data conversion that will be discussed in the next subsection. In this paper, we adopt the former as default. Fig. 3 depicts a line-to-area data conversion method based on the intersection topological relationship.

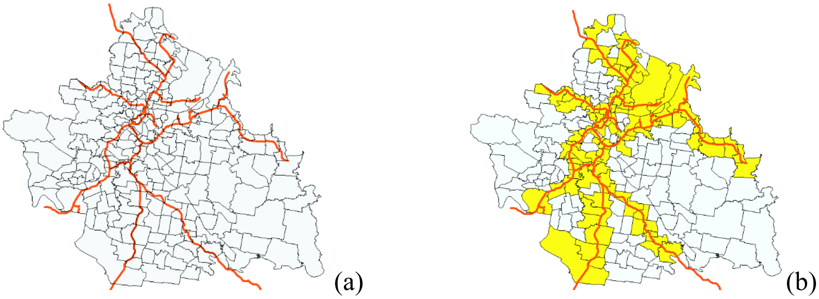


Fig. 3. Line-to-area data conversion: (a) A set of line data with the areal base map; (b) The base map with highlighted areal units representing the presence of the line data.

3.3.3 Area-to-Area Data Conversion

Intersection operation is still a solid candidate for this transformation. We adopt an approach somewhat similar to the areal-stealing interpolation technique [7]. Let $B = \{b_1, b_2,$□ $, b_n\}$ be an areal base map with a set n of areal units. Let $T = \{t_1, t_2,$□ $, t_m\}$ be an areal target map with a set m of areal units. Then, the target map is converted based on the following rules.

$$b_i.area = \sum_{j=1}^{n} b_i.IntersectionArea(t_j) \tag{1}$$

$$b_i.value = \sum_{j=1}^{n} (b_i.IntersectionArea(t_j) \times t_j.value)/b_i.area \tag{2}$$

Formula 1 denotes the area of an areal unit in the base map is the sum of its intersection areas with target areal units. Formula 2 represents an aggregate value of the area in the base map is the sum of productions of stolen areas and corresponding values in the target map. This is explained in Fig. 4. Fig. 4(a) displays an areal base map labeled with corresponding areas. Let us assume that an area highlighted, denoted by b_1, is of our interest. Fig. 4(b) shows a target areal map with different boundaries. It is labeled with the percentage of households owning more than two cars in the same study region. Our task is to aggregate the target layer based on the areal base map shown in Fig. 4(a). Fig. 4(c) depicts an overlay of b_1 and the target map labeled with area intersecting with areal units in the target map. As shown in Fig. 4(a), the area of b_1 is 16.34. According to Formula 1, it is the sum of stolen areas (3.37, 4.98, 6.15 and 1.84) shown in Fig. 4(c). The value of b_1 (b_1.*value*) can be computed as follows based on Formula 2: b_1.*value* = (3.37 * 0.27 + 4.98 * 0.24 + 6.15 * 0.51 + 1.84 * 0.67)/16.34 ≈ 40%. Thus, 40% of households in b_1 are regarded as owning more than two cars.

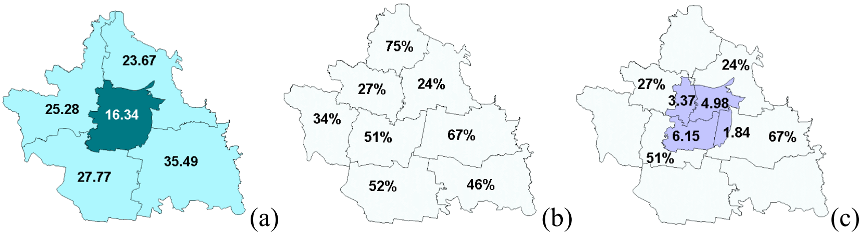


Fig. 4. Area-to-Area data conversion: (a) A base map labeled with corresponding areas; (b) A target map labeled with corresponding percentages of households owning more than two cars; (c) An overlay of the base map and the target map.

3.4 Categorization

GISs provide various built-in categorization schemes. ArcView GIS (version 3.1) offers *natural breaks*, *quantile*, *equal area*, *equal interval* and *standard deviations*. Natural breaks is the default categorization method in ArcView and this method identifies natural breakpoints between classes using a statistical formula called Jenk□s optimization [6]. This method is rather complex, but basically the Jenk□s method minimizes the sum of the variance within each of the classes. It is noted that natural breaks finds groupings and patterns inherent in data. Thus, it is chosen as default in our approach. In categorization tasks, the number of categories plays a critical role. Typically, 4-6 categories are recommended. Too small categories may lose details while too many categories cause confusion. In this paper, 5 is chosen as default for the number of groupings. However, users can explore a given dataset and they can choose the best categorization scheme for the certain dataset and can create a number of classes.

3.5 Mining Multivariate Associations

Apriori algorithm has been dominant in association rules mining since its first introduction in 1993 [1]. Many apriori-like algorithms [1,2,9,10] have been suggested using different indexing schemes or pruning techniques [9]. We use apriori algorithm in this paper since it is an influential and fundamental algorithm for finding frequent patterns. It employs an iterative level-wise search approach where *k*-items are used to explore (*k*+1)-items. It first finds frequent 1-items that satisfy the minimum support constraint. The sets are used to find frequent 2-items which will be used to find frequent 3-items. Non-monotonicity property, all nonempty subsets of a frequent item must also be frequent, is used to improve the efficiency of the level-wise generation of frequent items. Once frequent itemsets are generated, then minimum confidence is in play to find only strong rules. Readers can refer to the original paper [1,2] for details. In our approach, features correspond to items while areal units correspond to transactions. *Featuresets* denote sets of features. Apriori algorithm is used to find frequent featuresets and strong geospatial rules.

4 Case Study

Brisbane, the capital city of Queensland region of Australia, is continuously experiencing steady population growth [14,15]. Understanding of criminal activity in this region provides a valuable resource to city planners, policing agencies and criminologists. We select 217 urban suburbs of Brisbane as a study region. We use raw crime datasets recorded in the year of 1997 by Queensland Police Service. Obtained crime statistics have three main categories: □offences against the person□, □offences against property□and □other offences□. The first category □offences against the person□consists of subcategories: "homicide", "assaults", "sexual offences", "robbery", "extortion", "kidnapping" and "other offences against the person". The second category "offences against property" is composed of "breaking and entering", "arson", "other property damage", "motor vehicle theft", "stealing", "fraud" and "other offences against property". The third category "other offences" includes "drug offences", "prostitution", "liquor", "gaming offences", "trespassing" and "vagrancy", "good order offences", "traffic and related offences" and "miscellaneous offences". In addition, the subcategories could have several subsubcategories. For instances, "homicide" consists of "attempted murder", "conspiracy to murder", "driving causing death" and "manslaughter". The complex structure of crime dataset is not the only concern for crime activity analysis. Various geospatial features must be studied to identify salient features that cause criminal activities. In this experiment, a number of feature datasets including reserve, school, hospital, highway, caravan park, park, railway stations, convent, post office, police academy and airport including aerodromes are considered. All these datasets are aggregated onto an areal base map (217 suburbs) and categorized into several groupings using natural breaks. Most datasets are grouped into 5 categories: *very low*, *low*, *intermediate*, *high* and *very high*. Some datasets, scarcely recorded or line data types, are grouped into 2 categories: absence and presence. Some crime activities including gaming offences, prostitution and conspiracy to murder are not recorded in the study region in the year of 1997. Thus, they are ignored. For boolean association rules mining, groupings exhibiting *high*, *very high* and *intermediate* values are further grouped as hot spots indicating thriving criminal activities.

4.1 Time Complexity Analysis

Not surprisingly, small values of minimum support and minimum confidence tend to require more time in our test datasets since the number of frequent featuresets exponentially grows with smaller values. Experiments were performed on a 2.40GHz Pentium IV workstation with 1GB main memory. Time is measured in milliseconds. Fig. 5 shows time complexity analysis with the underlying crime dataset. Time is measured with 3

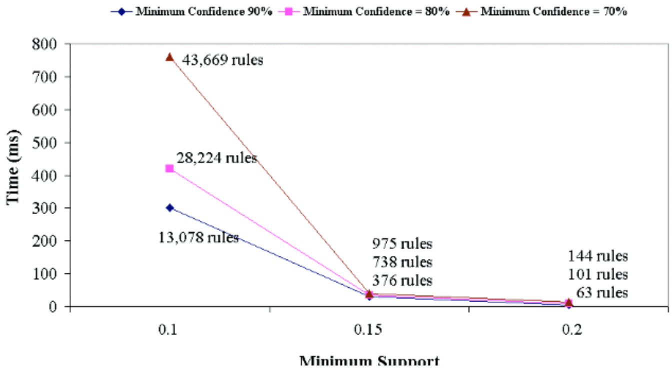


Fig. 5. Time complexity analysis with a real crime dataset and features recorded 1997 in Brisbane.

minimum support values (0.1, 0.15 and 0.2) with 3 different minimum confidence values (0.7, 0.8 and 0.9). All experiments are performed within a second as shown in this figure, which demonstrates the efficiency of association rules mining. This allows users to have prompt results and enables interactive and exploratory analysis.

4.2 Interesting Findings

Users can use must-contain constraints in association rules mining to tailor their search. They can put constraints on antecedent and/or consequent. The former (a must-contain featureset $\Rightarrow$ *) is used to explore possible consequences of the featureset. For instance, a city council wants to know all causal rules (casino $\Rightarrow$ *) that have casino in antecedent when it plans to introduce a new casino. If casino attracts too many criminal activities, then the council may not want to introduce one in the region. The latter (* $\Rightarrow$ a must-contain constraint) is used to explore possible stimuli of the featureset. These constraints were utilized in our experiments. Some interesting findings include:

- School highly likely implies occurrences of □unlawful entry with intent □dwelling□, □unlawful entry without violence □dwelling□and □other theft (excluding unlawful entry)□,
- Highway is implicative of □other stealing□, □other property damage□, □unlawful entry with intent □dwelling□, □unlawful entry without violence □dwelling□and □other theft (excluding unlawful entry)□,
- School and highway imply □motor vehicle theft□and □other theft (excluding unlawful entry)□,
- Parks exhibit a strong relationship with school,
- Railway stations attract □unlawful entry with intent □dwelling□and □unlawful entry without violence □dwelling□.

5 Concluding Remarks

This paper introduces a robust geospatial knowledge discovery framework. It requires two preprocessing steps: data conversion and categorization. Heterogeneous data types stored in multi-layered GISs are transformed into areal aggregates based on a base map. These numerical aggregates are categorized into several groupings to be mined. Experimental results are promising. It quickly reveals various interesting asymmetric causal relationships. These rules are valuable resources for city planning, geospatial decision-making, criminal precaution, regular police patrol and prediction.

This is a part of large project to investigate the analysis of criminal activities in relation to socio-economic, environmental and demographic factors in the Queensland area of Brisbane. The overall aim is to propose a robust framework that helps users mine frequent causal patterns within layer-based GIS environments. We use ArcView GIS for our experiments and use its scripting language AVENUE and dynamic link library to implement the three-phase geospatial mining framework.

References

1. Agrawal, R., Imielinski, R., Swami, A.N.: Mining Association Rules between Sets of Items inLarge Databases. In: Bunneman, P., Jajodia, S. (eds.): Proc. of the ACM Int. Conf. on Management of Data. ACM Press, Washington (1993) 207-216
2. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.): Proc. of the 20th Int. Conf. on Very Large Data Bases. Morgan Kaufmann Publishers, San Francisco (1994) 487-499
3. 3.http://www.informatik.unitrier.de/%7Eley/db/conf/sigmod/AgrawalS00.htmlAgrawal, R., Srikant, R.: Privacy-Preserving Data Mining. Proc. of the ACM Int. Conf. on Management of Data. ACM Press, Washington (200) 439-450
4. Bailey, T.C., Gatrell, A. C.: Interactive Spatial Analysis. Longman Scientific & Technical, Harlow UK (1995)
5. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.M.: Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. GeoInformatica (1997) 1(3):275-316
6. Dent, B. D.: Cartography: Thematic Map Design. WCB publishers, Dubuque USA (1996).
7. Gold, C.M.: Problems with Handling Spatial Data □the Voronoi Approach. CISM Journal (1991) 45:65-80
8. Estivill-Castro, V., Lee, I.: Data Mining Techniques for Autonomous Exploration of Large Volumes of Geo-referenced Crime Data. In: Pullar, D.V. (ed.): Proc. of the 6th Int. Conf. on Geocomputation (2001)
9. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2000)
10. Hipp, J., G□ntzer, U., Gholamareza,.: Algorithms for Association Rule Mining - A General Survey and Comparison. SIGKDD Explorations (2000) 2(1):58-64
11. Koperski, K., Han, J.: Discovery of Spatial Association Rules in Geographic Information Databases. In: Egenhofer, M.J., Herring, J.R. (eds.): Proc. of the 4th Int. Symp. on Large Spatial Databases. LNCS, Vol. 951. Springer-Verlag, Berlin Heidelberg (1995) 47-66
12. McHarg, I.L.: Design with Nature. Natural History Press, New York (1969)
13. Miller, H., Han, J.: Geographic Data Mining and Knowledge Discovery: An Overview. Cambridge University Press, Cambridge (2001)
14. Murray, A., Shyy, T.: Integrating Attribute and Space Characteristics in Choropleth Display and Spatial Data Mining. Int. J. of Geographic Information Science (2000) 14:649-667
15. Murray, A., McGuffog, I., Western, J., Mullins, P.: Exploratory Spatial Data Analysis Techniques for Examining Urban Crime. British J. of Criminology. (2001) 41L309-329
16. Roddick, J.F., Lees, B.G.: Paradigms for Spatial Data Warehousing for Geographic Knowledge Discovery. In: Miller, H.J., Han, J. (eds.): Geographic Data Mining and Knowledge Discovery: An Overview. Cambridge University Press, Cambridge (2001)
17. Shekhar, S., Huang, Y.: Discovering Spatial Co-location Patterns: A Summary of Results. In: Jensen, C.S., Schneider, M., Seeger, V.J., Tsotras, B. (eds.): Proc. of the 7th Int. Symp. on the Advances in Spatial and Temporal Databases. LNCS, Vol. 2121. Springer-Verlag, Berlin Heidelberg New York (2001) 236-256
18. Srikant, R., Agrawal, R.: Mining Quantitative Association Rules in Large Relational Tables. In: Jagadish, H.V., Mumick, I.S. (eds.): Pro. of the ACM Int. Conf. on Management of Data. ACM Press (1996) 1-12
19. Worboys, M.F.: GIS: A Computing Perspective. Taylor & Francis, London (1995)

Comparison between Objective Interestingness Measures and Real Human Interest in Medical Data Mining

Miho Ohsaki[1], Yoshinori Sato[1], Shinya Kitaguchi[1], Hideto Yokoi[2],
and Takahira Yamaguchi[1]

[1] Shizuoka University, Faculty of Information
3-5-1 Johoku, Hamamatsu-shi, Shizuoka 432-8011, Japan
{miho, cs8037, cs9026, yamaguti}@cs.inf.shizuoka.ac.jp
http://www.cs.inf.shizuoka.ac.jp/~miho/
http://panda.cs.inf.shizuoka.ac.jp/
[2] Chiba University Hospital, Medical Informatics
1-8-1 Inohana, Chuo-ku, Chiba-shi, Chiba 260-0856, Japan
yokoih@telemed.ho.chiba-u.ac.jp

Abstract. This research empirically investigates the performance of conventional rule interestingness measures and discusses their availability to supporting KDD through system-human interaction in medical domain. We compared the evaluation results by a medical expert and that by selected measures for the rules discovered from a dataset on hepatitis. Recall and χ^2 Measure 1 demonstrated the highest performance, and all measures showed different trends under our experimental conditions. These results indicated that some measures can predict really interesting rules at a certain level and that their combinational use in system-human interaction will be useful.

1 Introduction

This research has two backgrounds: the necessity of system-human interaction support in medical Knowledge Discovery in Databases (KDD) and the problem of conventional measures of rule interestingness. On the former background, medical data mining is one of active research fields in KDD due to its scientific and social contribution. We have been conducted case studies on hepatitis and repeated obtaining rules to predict prognosis from a clinical dataset and their evaluation from a medical expert, improving the mining conditions. This process made us recognize the significance of system-human interaction to enhance rule quality by reflecting the domain knowledge and the requirement of a medical expert and raised the issue on its semi-automatic support [1]. On the latter background, rule interestingness is also an important field in KDD, and there have been many studies to formulate interestingness measures and evaluate rules with them instead of humans. Some latest studies made a survey on individually proposed objective measures and tried to categorize and/or analyze them [2][3][4]. However, they were mainly based on mathematical analysis, and little attention has been given to how much objective measures can reflect real human interest.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1072-1081, 2004.

Therefore, the purposes of this research are (1) investigating the conventional interestingness measures and comparing them with the rule evaluation results by a medical expert, and (2) discussing whether they are useful to support system-human interaction in medical KDD. In this paper, Section 2 introduces conventional interestingness measures and selects several measures suitable to our purpose. Section 3 shows the experiment to evaluate rules on chronic hepatitis with the measures and to compare the evaluation results by them with that by a medical expert. In addition, it discusses the availability of the measures for system-human interaction support. Finally, Section 4 concludes the paper and comments on the future work.

2 Related Work

2.1 Our Previous Research on Medical Data Mining

We have conducted case studies [1] to discover the rules predicting prognosis based on diagnosis from a dataset of the medical test results on viral chronic hepatitis [5]. The set of rule generation by our mining system and rule evaluation by a medical expert was iterated two times and leaded us to discover the rules valued as interesting ones by the medical expert. At first, we finely pre-processed the dataset based on medical expert's advice since such a real medical dataset is ill-defined and has many noises and missing values. We then performed a popular time-series mining technique [6], from which we removed a subsequence extraction process to avoid STS clustering problem [7], on the pre-processed dataset. We extracted the representative temporal patterns from the dataset by clustering and generated the rules consisting of the patterns by a decision tree.

The left graph in Fig. 1 shows one of the rules, which the medical expert focused on, obtained in the first mining. It estimates the future trend of GPT, one of major medical tests to grasp chronic hepatitis symptom, in the future one year by using the change of several medical test results in the past two years. The medical expert commented on it as follows: the rule offers a hypothesis that GPT value changes with about a three-years cyclic, and the hypothesis is interesting since it differs from the conventional common sense of medical experts that GPT value basically decreases in a monotone. We then improved our mining system, extended the observation term, and generated new rules. The right graph in Fig. 1 shows one of the rules, which the medical expert valued, obtained in our second mining. The medical expert commented on it that it implies GPT value globally changes two times in the past five years and more strongly supports the hypothesis of GPT's cyclic change.

As the next step, we tried to systematize the knowledge obtained through the iteration of mining and evaluating process on the system-human interaction to polish up rules. We formulated its concept model that describes the roles and the functions of a mining system and a human user (See Fig. 2) and the framework to semi-automatically support system-human interaction based on the model (See Fig. 3) [8].

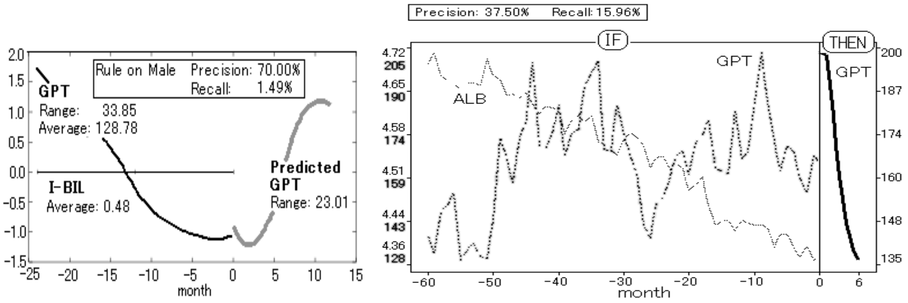


Fig. 1. Rules valued by a medical expert in the first mining (left) and the second mining (right).

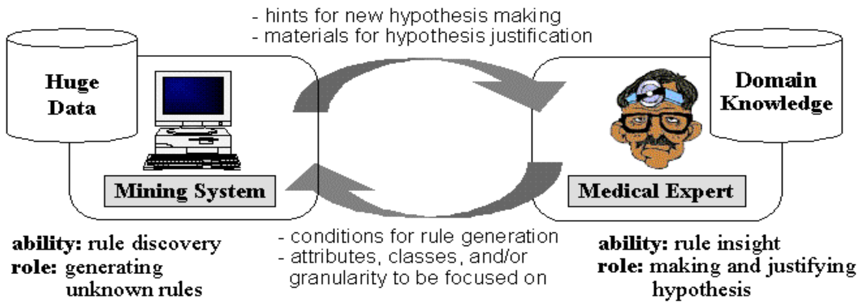

- hints for new hypothesis making
- materials for hypothesis justification
Huge Data
Mining System
Domain Knowledge
Medical Expert
ability: rule discovery
role: generating unknown rules
- conditions for rule generation
- attributes, classes, and/or granularity to be focused on
ability: rule insight
role: making and justifying hypothesis

Fig. 2. Interaction model between a mining system and a human expert at a concept level.

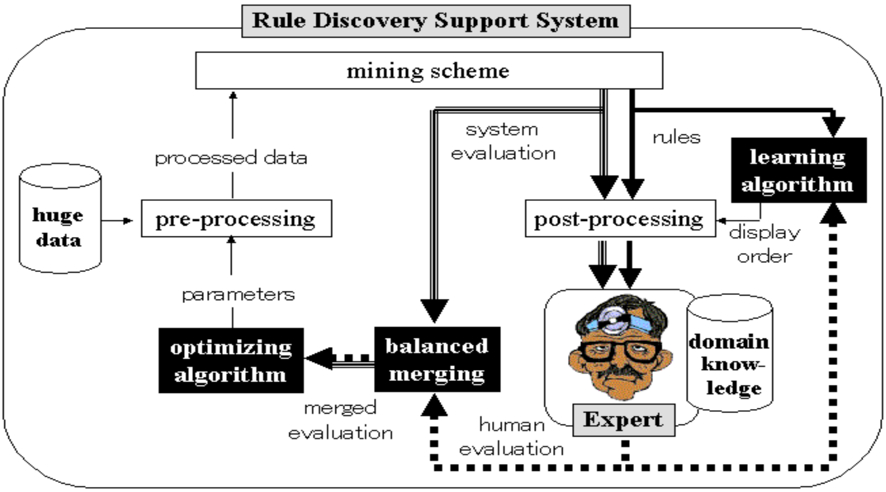


Fig. 3. Framework of the semi-automatic support of system-human interaction.

As shown in Fig. 2, a mining system discovers the rules faithfully to the data and offers them to a medical expert as the materials for hypothesis generation and justification. (Note that the word 'justification' in this paper does not mean the highly reliable proof of a hypothesis by additional medical experiments under strictly controlled conditions. It means the additional information extraction from the same data to enhance the reliability of an initial hypothesis.) While, the medical expert generates and justifies a hypothesis, namely a seed of new knowledge, by evaluating the rules based on his/her domain knowledge. A system to support such interaction requires the function to generate and present rules to a medical expert based on the validity of rules at the viewpoints of mathematical features of data and subjective human evaluation criteria. The flow of "System Evaluation" and "Human Evaluation" in Fig. 3 describes that.

Our previous research notified us that it is required for realizing the framework in Fig. 3 to investigate whether the rule interestingness measures are useful for "System Evaluation" and how is the relation between "System Evaluation" and "Human Evaluation". Therefore, this research selects several conventional objective measures and compares the rule evaluation results by them with that by a medical expert [8].

2.2 Conventional Rule Interestingness Measures

Interestingness measures are categorized into objective and subjective ones [3][4]. Objective measures mean how a rule is mathematically meaningful based on the distribution structure of the instances related to the rule. They are mainly used to remove meaningless rules rather than to discover really interesting ones for a human user, since they do not include domain knowledge [9]-[19]. On the other hand, subjective measures mean how a rule fits with a belief, a bias, or a rule template formulated beforehand by a human user. They are useful to discover really interesting rules to some extend due to their built-in domain knowledge. However, they depend on the precondition that a human user can clearly formulate his/her own interest and do not discover absolutely unexpected knowledge [20]-[26]. Few subjective measures adaptively learn real human interest through system-human interaction [27].

The conventional interestingness measures, not only objective but also subjective, do not directly reflect the interest that a human user really has. To avoid the confusion of real human interest, objective measure, and subjective measure, we clearly differentiate them in the next paragraph. While we define "Real Human Interest" by ourselves, the definitions of interestingness measures are based on many conventional studies on interestingness measures [2]-[4][9]-[28].

Objective Measure. The feature such as correctness, uniqueness, and strength of a rule or a set of rules, calculated by mathematically analyzing data structure. It does not include human evaluation criteria.

Subjective Measure. The similarity or the difference between the information on interestingness beforehand given by a human user and that obtained from a rule or a set of rules. Although it includes human evaluation criteria in its initial state, its calculation of similarity or difference is mainly based on the mathematical analysis of data structure.

Real Human Interest. The interest which a human user really feels for a rule in his/her mind. It is formed by the synthesis of cognition, domain knowledge, individual experiences, and the influences of the rules that he/she evaluated before.

Here we introduce conventional representative objective measures, because this research mainly focuses on the relation between them and real human interest. They can be categorized into some groups using evaluation criterion, evaluation target, and theory for analysis. The evaluation target means whether an objective measure evaluates a rule or a set of rules. This research does not deal with the objective measures for a set of rules but ones for a rule, because we focus on the quality of each rule.

Table 1. The list of the objective measures of rule interestingness. The measures used in this research are Bold-styled in the first column. "T. F." in the second column means theoretical foundation as follows. N: Number of instances included in antecedent and/or consequent. P: Probability of antecedent and/or consequent. S: Statistical variable based on P. I: Information of antecedent and/or consequent. D: Distance of a rule from the others based on rule attributes. C: Complexity of the tree structure of a rule.

Measure Name	T. F.	Evaluation Criterion
Rule Interest [9]	N	Dependency between the antecedent and the consequent of a rule
Support	P	Generality of a rule
Precision	P	Performance of a rule to predict its consequent
Recall	P	Performance of a rule not to leak its consequent
Accuracy	P	Summation of precision and its converse of contraposition
Lift	P	Dependency between the antecedent and the consequent of a rule
Leverage	P	Dependency between the antecedent and the consequent of a rule
Reliable Exceptions [16]	P	A rule with small support and high precision
GOI [19]	P	Multiplication of the support and the antecedent-consequent dependency of a rule
Surprisingness [15]	P	A rule occurring Sympson's paradox
χ^2 Measure 1 [14]	S	Dependency between the antecedent and the consequent of a rule
χ^2 Measure 2 [14]	S	Similarity between two rules
J-Measure [10]	I	Multiplication of the coverage and the antecedent-consequent dependency of a rule
General Measure [17]	S , I	Fusion of χ^2 Measure 1 and information gain measure
Distance Metric [13]	D	Distance of a rule from the other rule with highest coverage
DLI [12]	D	Distance of a rule from the other rules
Peculiarity [18]	D	Distance of the attribute value of a rule from frequent attribute values
I-Measure [11]	C	Complexity of a rule

Table 1 shows some major objective measures. They assume one of the following evaluation criteria and examine how a rule matches with the criteria by calculating the instance distribution difference between data and a rule or between the antecedent and the consequent of a rule. **Correctness:** How many instances the antecedent and/or the consequent of a rule supports, or how strong their dependence is [9][10][14][17]. **Information Richness:** How much information a rule possesses [11]. **Generality:** How similar the trend of a rule is to that of all data [13]. **Uniqueness:** How different the trend of a rule is from that of all data [12][15][18] or the other rules [13][14].

Objective measures are useful to automatically remove obviously meaningless rules. However, some evaluation criteria have the contradiction to each other such as generality and uniqueness, and the evaluation criterion of an objective measure may not match with or may contradict real human interest. For example, a rule with a plenty of information may be too complex for a human user to understand. Although the validity of objective measures has been mathematically proven and/or experimentally discussed using some benchmark data, very few attempts have been made at the total comparison of them and the investigation of the relation between them and real human interest for a concrete application.

In a sense, it may be proper not to investigate the relation between objective measures and real human interest, since their evaluation criteria do not include knowledge on rule semantics and are obviously not the same of humans. However, our idea is that they may be useful to support the KDD through system-human interaction if they possess a certain level of performance to detect really interesting rules. In addition, they may offer a human user unexpected new viewpoints. That is the motivation of this research.

In this research we selected the most popular ones (Support, Precision, Recall, Lift, and Leverage), probability-based one (GOI), statistics-based one (χ^2 measure 1), and information-based one (J-Measure) from the objective measures in Table 1, and used them in the experiment in Section 3.

3 Experiment to Compare Objective Interestingness Measures with Real Human Interest

3.1 Experimental Conditions

We used two sets of rules and their evaluation results by a medical expert obtained through mining process iteration in our previous research (Refer Section 2.1). After each mining, the medical expert evaluated the rules and gave each rule one of the following rule quality labels: Especially-Interesting (**EI**), Interesting (**I**), Not-Understandable (**NU**), and Not-Interesting (**NI**). **EI** means that the rule was a key factor to generate the hypothesis of GPT's cyclic change in the first mining or to justify it in the second mining. As the results, we obtained 12 and 8 interesting rules, which were labeled with **EI** or **I**, in the first and the second mining, respectively.

We then applied the objective measures selected in Section 2.2 to the same rules and sorted them in the descending order of their evaluation values. Referring the number of rules labeled with **EI** by the medical expert, we gave the label **EI** to the top three rules in the first mining and the top two in the second. Similarly, we gave the label **I** to the top 9 rules excluding ones labeled with **EI** in the first mining and the top 6 in the second.

3.2 Results and Discussion

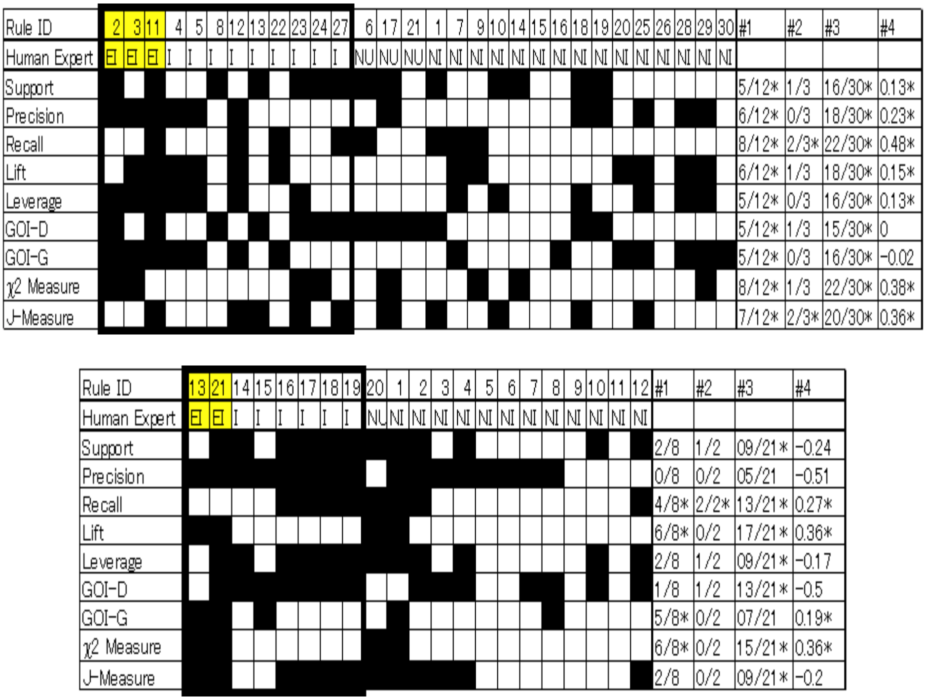
The upper and the lower tables in Fig. 4 show the evaluation results in the first and the second mining, respectively. They describe how the evaluation results of an objective measure matched with that of a medical expert. For each objective measure, the more the number of white cells in the left side of a table, the better its performance to estimate real human interest. Note that there are two types of GOI, GOI-D (GOI emphasizing Dependency) and GOI-G (GOI emphasizing Generality) in Fig. 4. GOI is the multiplication of antecedent-consequent dependency factor and generality one and possesses a parameter to balance them. The value of dependency factor was double that of generality one for GOI-D and vice versa for GOI-G.

To clearly grasp the trend of the experimental results, we define the comprehensive criteria to evaluate the performance of an objective measure: **#1**: Performance on **I** (the number of rules labeled with **I** by the objective measure over that by the medical expert). **#2**: Performance on **EI** (the number of rules labeled with **EI** by the objective measure over that by the medical expert). **#3**: Number-based performance on all evaluation (the number of rules with the same evaluation results by the objective measure and the medical expert over that of all rules). **#4**: Correlation-based performance on all evaluation (the correlation coefficient between the evaluation results by the objective measure and that by the medical expert). The results of analysis based on these criteria are shown in the right side of the tables in Fig. 4. The symbol '*' besides a value in the right side cells means that the value is greater than that in case rules are randomly selected as EI or I. Therefore, an objective measure with '*' has at least higher performance than random selection.

At first, we discuss the results in each mining. In the first mining, Recall and χ^2 Measure 1 demonstrated the highest performance, and Support, Leverage, GOI-G, and GOI-D demonstrated the lowest (See the upper table in Fig. 4). In the second mining, Lift and χ^2 Measure 1 demonstrated the highest performance, and Precision demonstrated the lowest (See the lower table in Fig. 4). Next, we discuss the whole trend of the results through the first and second mining. Recall and χ^2 Measure 1 maintained the highest performance, and Support and Leverage the lowest. The other objective measures maintained or slightly changed their middle performance. Although the objective measures with the highest or the second highest performance failed to detect some of EI or I rules, their usefulness to predict real human interest was confirmed at a certain level, since they had comparatively many white cells and '*' for all comprehensive criteria.

We consider why such trends appeared comparing them with the analysis of medical expert's comments on evaluation. The analysis illustrated the following points of medical expert's observation: (1) the medical expert focused on the whole shape of temporal patterns in a rule rather than the rule performance to predict prognosis, (2) he evaluated a rule totally considering reliability, unexpectedness, difference between rules, and so on, (3) although reliability was one of important evaluation factors, many reliable rules were not interesting due to their well-knownness.

It is inferred that (1) caused the highest performance of χ^2 Measure 1. Only χ^2 Measure 1 uses the instances for the all combinations of supporting antecedent, not supporting antecedent, supporting consequent, and not supporting consequent [14]. Accordingly, it valued the rules in which the temporal patterns in the antecedent and that in the consequent were smoothly connected. This feature of χ^2 Measure 1 possibly met the medical expert's needs. The lowest performance of Support seems to be deserved with considering (3). We could not yet estimate the reason of the highest performance of Recall and the lowest performance of Leverage based on the comments.



Rule ID	2	3	11	4	5	8	12	13	22	23	24	27	6	17	21	1	7	9	10	14	15	16	18	19	20	25	26	28	29	30	#1	#2	#3	#4
Human Expert	EI	EI	EI	I	I	I	I	I	I	I	I	I	NU	NU	NU	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI				
Support	■	□	■	□	□	■	□	■	□	■	■	■	■	■	□	■	□	□	■	■	□	□	■	■	□	□	□	□	□	□	5/12*	1/3	16/30*	0.13*
Precision	■	■	■	■	■	□	■	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	■	■	□	■	□	■	■	□	6/12*	0/3	18/30*	0.23*
Recall	□	□	■	□	□	□	■	□	■	□	□	■	■	□	□	■	■	■	□	□	□	□	□	□	□	□	□	□	□	□	8/12*	2/3*	22/30*	0.48*
Lift	□	■	■	■	■	□	■	□	■	□	□	□	□	□	□	□	■	■	□	□	□	□	□	□	■	■	□	■	■	□	6/12*	1/3	18/30*	0.15*
Leverage	■	■	■	■	■	□	■	□	□	■	□	□	□	□	□	□	■	□	■	□	□	□	■	□	□	■	□	■	■	□	5/12*	0/3	16/30*	0.13*
GOI-D	■	□	■	□	□	■	□	■	□	■	■	■	■	■	■	■	□	□	□	□	□	□	■	■	□	□	□	□	□	□	5/12*	1/3	15/30*	0
GOI-G	■	■	■	■	■	□	■	□	■	□	□	□	□	□	□	□	■	□	□	□	□	■	□	□	■	■	□	■	■	■	5/12*	0/3	16/30*	-0.02
χ2 Measure	■	■	□	□	□	□	□	□	□	■	■	□	□	■	□	□	□	■	□	■	□	□	□	□	□	□	□	□	■	□	8/12*	1/3	22/30*	0.38*
J-Measure	□	□	■	□	□	□	■	■	□	■	□	■	□	■	□	■	□	□	■	□	□	□	■	□	□	■	□	□	□	□	7/12*	2/3*	20/30*	0.36*

Rule ID	13	21	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10	11	12	#1	#2	#3	#4
Human Expert	EI	EI	I	I	I	I	I	I	NU	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI				
Support	□	■	■	□	■	■	■	■	■	■	■	□	■	□	□	□	□	□	■	□	■	2/8	1/2	09/21*	-0.24
Precision	■	■	■	■	■	■	■	■	□	■	■	■	■	■	■	■	■	□	□	□	□	0/8	0/2	05/21	-0.51
Recall	□	□	□	□	■	■	■	■	■	■	■	□	□	□	□	□	□	□	□	□	■	4/8*	2/2*	13/21*	0.27*
Lift	■	■	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□	□	□	□	6/8*	0/2	17/21*	0.36*
Leverage	□	■	■	□	■	■	■	■	■	■	■	□	■	□	□	□	□	□	■	□	■	2/8	1/2	09/21*	-0.17
GOI-D	□	■	■	■	■	■	■	■	□	□	■	■	■	□	□	■	■	□	■	□	■	1/8	1/2	13/21*	-0.5
GOI-G	■	■	□	■	□	□	□	□	□	■	□	□	□	□	□	□	■	□	□	□	□	5/8*	0/2	07/21	0.19*
χ2 Measure	■	■	□	□	□	□	□	□	■	■	□	□	□	□	□	□	□	□	□	□	□	6/8*	0/2	15/21*	0.36*
J-Measure	■	■	□	□	■	■	■	■	■	■	■	■	■	□	□	□	□	□	□	□	■	2/8	0/2	09/21*	-0.2

Fig. 4. Evaluation results by a medical expert and the selected objective measures for the rules obtained in the first mining (upper) and that in the second mining (lower). Each line means a set of evaluation results by the medical expert or the objective measure, and each column means each rule. The rules are sorted in the descending order of the evaluation values given by the medical expert. A square in the left side surrounds the rules labeled with **EI** or **I** by the medical expert. For each objective measure, white and black cells mean that the evaluation of the objective measure was the same of the medical expert and that it was not, respectively. The four columns in the right side show the performance on each comprehensive criterion. '*' means the value is greater than that in case rules are randomly selected as **EI** or **I**.

The following implied that the composition of several objective measures may accomplish higher performance: the expert's comment (2) and the mosaic-like patterns of white and black cells in Fig.4. It is possible to formulate a function consisting of the summation of weighted outputs from different objective measures. We can also learn a decision tree using these outputs as attributes and the evaluation result by the medical expert as a class. If we obtain such a function or a decision tree, it explains real human interest reductively and will be useful to recommend interesting rule candidates by predicting their evaluation results through system-human interaction shown in Fig.2.

4 Conclusions and Future Work

This research discussed how objective measures can contribute to detect interesting rules for a medical expert through the experiment using a real hepatitis dataset. Recall and χ^2 Measure 1 demonstrated good performance, and the objective measures used in this research had compensatory relationship for each other. It was indicated that their combination will be useful to support system-human interaction. Now, we continue other experiments using 43 objective measures, two medical datasets, and the rule evaluation results by two medical experts in the same way. In addition to that, we will formulate the composition function of objective measures and apply it to the semi-automatic support of system-human interaction in medical KDD.

References

1. Ohsaki, M., Sato, Y., Yokoi, H., Yamaguchi, T.: A Rule Discovery Support System for Sequential Medical Data, – In the Case Study of a Chronic Hepatitis Dataset –. Proceedings of International Workshop on Active Mining AM-2002 in IEEE International Conference on Data Mining ICDM-2002 (2002) 97–102
2. Hilderman, R. J., Hamilton, H. J.: Knowledge Discovery and Measure of Interest. Kluwer Academic Publishers (2001)
3. Tan, P. N., Kumar V., and Srivastava, J.: Selecting the Right Interestingness Measure for Association Patterns. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-2002 (2002) 32-41
4. Yao, Y. Y. and Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD-1999 (1999) 479-488
5. Hepatitis Dataset for Discovery Challenge. in Web Page of European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD-2002 (2002) http://lisp.vse.cz/challenge/ecmlpkdd2002/index.html
6. Das, G., King-Ip, L., Heikki, M., Renganathan, G., Smyth, P.: Rule Discovery from Time Series. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-1998 (1998) 16–22
7. Lin, J., Keogh, E., and Truppel, W.: (Not) Finding Rules in Time Series: A Surprising Result with Implications for Previous and Future Research. Proceedings of International Conference on Artificial Intelligence IC-AI-2003 (2003) 55–61
8. Ohsaki, M., Sato, Y., Yokoi, H., Yamaguchi, T.: Investigation of Rule Interestingness Measures in Medical Data Mining. Proceedings of International Workshop on Active Mining AM-2003 in International Symposium on Methodologies for Intelligent Systems ISMIS-2003 (2003) 85–97
9. Piatetsky-Shapiro, G.: Discovery, Analysis and Presentation of Strong Rules. in Piatetsky-Shapiro, G., Frawley, W. J. (eds.): Knowledge Discovery in Databases. AAAI/MIT Press (1991) 229–248

10. Smyth, P., Goodman, R. M.: Rule Induction using Information Theory. in Piatetsky-Shapiro, G., Frawley, W. J. (eds.): Knowledge Discovery in Databases. AAAI/MIT Press (1991) 159–176
11. Hamilton, H. J., Fudger, D. F.: Estimating DBLearn's Potential for Knowledge Discovery in Databases. Computational Intelligence, 11, 2 (1995) 280–296
12. Dong, G., Li, J.: Interestingness of Discovered Association Rules in Terms of Neighborhood-Based Unexpectedness. Proceedings of Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD-1998 (1998) 72–86
13. Gago, P., Bento, C.: A Metric for Selection of the Most Promising Rules. Proceedings of European Conference on the Principles of Data Mining and Knowledge Discovery PKDD-1998 (1998) 19–27
14. Morimoto, Y., Fukuda, T., Matsuzawa, H., Tokuyama, T., Yoda, K.: Algorithms for Mining Association Rules for Binary Segmentations of Huge Categorical Databases. Proceedings of International Conference on Very Large Databases VLDB-1998 (1998) 380–391
15. Freitas, A. A.: On Rule Interestingness Measures. Knowledge-Based Systems, 12, 5–6 (1999) 309–315
16. Liu, H., Lu, H., Feng, L., Hussain, F.: Efficient Search of Reliable Exceptions. Proceedings of Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD-1999 (1999) 194–203
17. Jaroszewicz, S., Simovici, D. A.: A General Measure of Rule Interestingness. Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery PKDD-2001 (2001) 253–265
18. Zhong, N., Yao, Y. Y., Ohshima, M.: Peculiarity Oriented Multi-Database Mining. IEEE Transaction on Knowledge and Data Engineering, 15, 4 (2003) 952–960
19. Gray, B., Orlowska, M. E.: CCAIIA: Clustering Categorical Attributes into Interesting Association Rules. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD-1998 (1998) 132–143
20. Klementtinen, M., Mannila, H., Ronkainen, P., Toivone, H., Verkamo, A. I.: Finding Interesting Rules from Large Sets of Discovered Association Rules. Proceedings of International Conference on Information and Knowledge Management CIKM-1994 (1994) 401–407
21. Kamber, M., Shinghal, R.: Evaluating the Interestingness of Characteristic Rules. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-1996 (1996) 263–266
22. Liu, B., Hsu, W., Chen, S., Mia, Y.: Analyzing the Subjective Interestingness of Association Rules. Intelligent Systems, 15, 5 (2000) 47–55
23. Liu, B., Hsu, W., Mia, Y.: Identifying Non-Actionable Association Rules. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-2001 (2001) 329–334
24. Padmanabhan, B., Tuzhilin, A.: A Belief-Driven Method for Discovering Unexpected Patterns. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-1998 (1998) 94–100
25. Sahara, S.: On Incorporating Subjective Interestingness into the Mining Process. Proceedings of IEEE International Conference on Data Mining ICDM-2002 (2002) 681–684
26. Silberschatz, A., Tuzhilin, A.: On Subjective Measures of Interestingness in Knowledge Discovery. Proceedings of International Conference on Knowledge Discovery and Data Mining KDD-1995 (1995) 275–281
27. Terano, T., Inada, M.: Data Mining from Clinical Data using Interactive Evolutionary Computation: in Ghosh, A., Tsutsui, S. (eds.): Advances in Evolutionary Computing. Springer (2003) 847–862
28. Hamilton, H. J., Shan, N., Ziarko, W.: Machine Learning of Credible Classifications. Proceedings of Australian Conference on Artificial Intelligence AI-1997 (1997) 330–339

Boosting with Data Generation: Improving the Classification of Hard to Learn Examples

Hongyu Guo and Herna L Viktor

School of Information Technology and Engineering, University of Ottawa
800 King Edward St., Ottawa, Ontario, Canada, K1N 6N5
{hguo028, hlviktor}@site.uottawa.ca

Abstract. An ensemble of classifiers consists of a set of individually trained classifiers whose predictions are combined to classify new instances. In particular, boosting is an ensemble method where the performance of weak classifiers is improved by focusing on "hard examples" which are difficult to classify. Recent studies have indicated that boosting algorithm is applicable to a broad spectrum of problems with great success. However, boosting algorithms frequently suffer from over-emphasizing the hard examples, leading to poor training and test set accuracies. Also, the knowledge acquired from such hard examples may be insufficient to improve the overall accuracy of the ensemble. This paper describes a new algorithm to solve the above-mentioned problems through data generation. In the DataBoost method, hard examples are identified during each of the iterations of the boosting algorithm. Subsequently, the hard examples are used to generate synthetic training data. These synthetic examples are added to the original training set and are used for further training. The paper shows the results of this approach against ten data sets, using both decision trees and neural networks as base classifiers. The experiments show promising results, in terms of the overall accuracy obtained.

1 Introduction

An ensemble consists of a set of individually trained classifiers whose predictions are combined when classifying novel instances. Ensemble approaches are an active area of research, with a number of research groups investigating techniques for combining the predictions of multiple classifiers to produce a single ensemble model [1, 2, 4]. This interest is due to the fact that, in complex to learn domain, the resulting model is often more accurate than the results of individual algorithms [4, 5, 6]. One of the most popular and widely applicable methods for creating accurate ensembles of classifiers is boosting [5, 7, 8]. Boosting is a general method which attempts to "boost" the accuracy the learning algorithm through focusing on "hard to learn" examples [18]. Boosting algorithms tend to generate distributions that concentrate on examples which are hard to be correctly classified, thus challenging the weak learning algorithm to perform well on these harder parts of the sample space [4]. However, previous work has demonstrated that boosting frequently suffer from over-emphasizing the hard training examples [5, 17, 18], leading to a decrease in predictive accuracy. Moreover, insufficient knowledge about the hard examples from the original training

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1082-1091, 2004.

data set often prevent boosting algorithms from obtaining large performance gains [16, 18].

This paper discusses an approach to assist boosting avoid the above-mentioned problems. In the DataBoost method, we identify hard examples from the training data. Next, synthetic training examples based on the hard examples are generated and added to the original training data to train the component classifiers. This prevents boosting from over-emphasizing the hard data in the training data set, thus reducing the boosting error. Moreover, with additional synthetic data, we can provide complementary knowledge about the hard examples on which boosting algorithms need to concentrate in the next iteration. The additional knowledge is subsequently used to improve the performance in terms of predictive accuracy.

The paper is organized as follows. Section 2 introduces ensemble-based algorithms based on boosting. This is followed, in Section 3, with a description of the DataBoost algorithm which combines boosting and data generation. The section discusses how hard examples are selected and how synthetic data are generated. Section 4 evaluates the DataBoost algorithm against ten data sets from the UCI data set repository [16], using both C4.5 decision trees [9] and Neural Networks as base classifiers [20].

Table 1. Hypothetical runs of Boosting. Assume there are five training examples. Assume example 1 is an "outlier" and is hard for the component learning algorithm to classify correctly. The Boosting uses all of the examples from the original training sets, but the hard example (example 1) is getting more weight in later training sets, which forces the Boosting algorithm to concentrate on correctly predicting it.

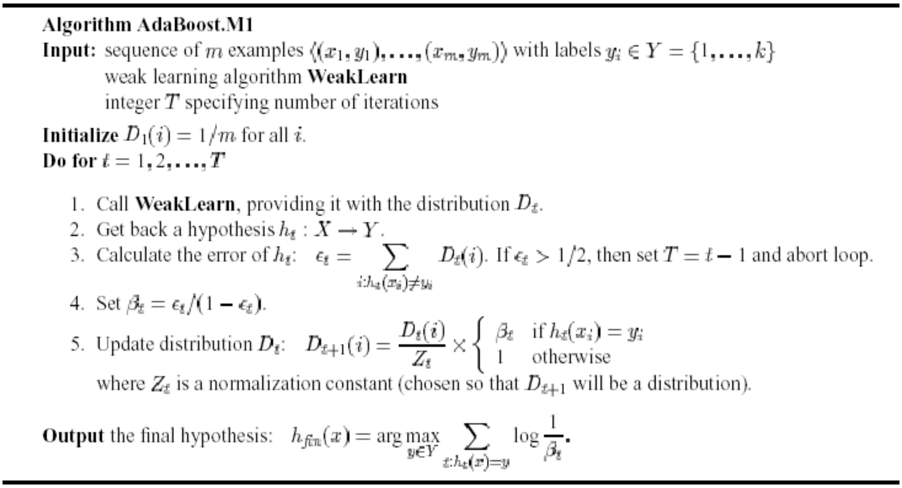
A sample of a single classifier on an imaginary set of data	
	(Original) Training Set
Training-set-1 and their weights:	1(0.20), 2(0.20), 3(0.20), 4(0.20), 5(0.20)

A sample of Boosting on the same examples	
	(Re-weighted) Training Set
Training-set-1 and their weights:	1(0.200), 2(0.200), 3(0.200), 4(0.20), 5(0.20)
Training-set-2 and their weights:	1(0.3), 2(0.175), 3(0.175), 4(0.175), 5(0.175)
Training-set-3 and their weights:	1(0.5), 2(0.125), 3(0.125), 4(0.125), 5(0.125)
Training-set-4 and their weights:	1(0.80), 2(0.050), 3(0.050), 4(0.050), 5(0.05)

2 Boosting Classifiers

Boosting encompasses a family of methods which focuses on the production of a series of classifiers. The outputs of these classifiers are combined using weighted voting in the final prediction of the model [5]. Boosting reduces the error in the variance term [3, 10] and attempts to reduce the error in the bias term as well [4]. It is this capability that makes boosting an appropriate algorithm for combining the predictions of "weak" learning algorithms [7]. In each step of the series, the training examples are re-weighted and chosen based on the performance of earlier classifiers in the training series. This produces a set of "easy" examples with low weight, and a set of hard ones with high weight. During each of the iterations, boosting attempts to produce new classifiers that are better able to predict examples for which the precious classifier's performance is poor. This is achieved by concentrating on classifying the hard examples correctly.

Table 1 shows a hypothetical run of boosting with re-weighting the original data set. In this figure, instance 1 is a hard example that the initial classifiers tend to misclassify. When using the second training set, example 1 has more weight since it was misclassified by the first classifiers. For the final training set, example 1 bears predominant weight; thus, the final component model will concentrate on classifying this example correctly, and the overall test-set error for this classifier should subsequently become very high.



```
Algorithm AdaBoost.M1
Input: sequence of $m$ examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ with labels $y_i \in Y = \{1, \ldots, k\}$
       weak learning algorithm WeakLearn
       integer $T$ specifying number of iterations
Initialize $D_1(i) = 1/m$ for all $i$.
Do for $t = 1, 2, \ldots, T$
    1. Call WeakLearn, providing it with the distribution $D_t$.
    2. Get back a hypothesis $h_t : X \to Y$.
    3. Calculate the error of $h_t$: $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$. If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop.
    4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.
    5. Update distribution $D_t$: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$
       where $Z_t$ is a normalization constant (chosen so that $D_{t+1}$ will be a distribution).
Output the final hypothesis: $h_{fin}(x) = \arg\max_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$.
```

Fig. 1. The AdaBoost algorithm [4]

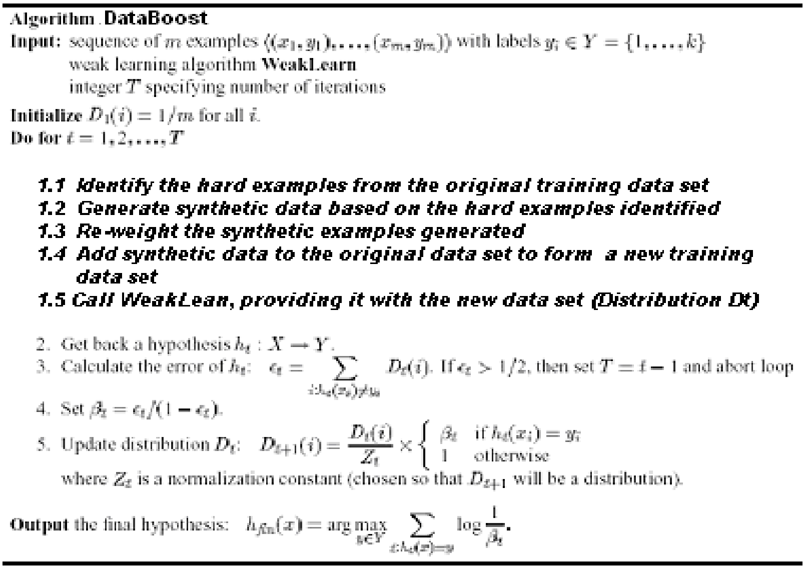
The AdaBoost algorithm, as shown in Figure 1, was introduced in 1995 by Freund and Schapire [19]. The AdaBoost algorithm is widely used, since it is fast, robust, simple and easy to program. It has no parameters to tune, requires no prior knowledge about the weak classifier and may be flexibly combined with any method for finding weak hypotheses [16]. As depicted in the figure, the final AdaBoost classifier consists of a weighted average of all the weak classifiers. The algorithm proceeds as follows. The initial weights of all examples are set to be *1/m*. After each trained classifier is added to the ensemble, the weights of examples being correctly classified by the current trained classifier are left unchanged, and otherwise the weights are increased by multiply by a factor. The weights are then renormalized by dividing by the normalization constant. Thus 'easy' examples that are correctly classified by many of the previous trained classifiers get lower weight, and hard examples which tend often to be misclassified get higher weight. The boosting algorithm concentrates on the harder examples, and tends challenging the weak learning algorithm to perform well on these harder parts of the sample space. Due to this property, boosting easily over-emphasizes the hard examples. Inadequate knowledge about the hard examples, which the next iteration of the boosting need to focus on, will prevent boosting from obtaining large performance gains as well.

3 Boosting with Data Generation

In this section, we describe a new variation of Boosting, the so-called DataBoost algorithm, which uses synthetic data generated based on the hard examples to improve the performance of boosting. The DataBoost algorithm uses synthetic data to prevent boosting algorithms from over-emphasizing the hard examples. These examples thus provide complementary knowledge about hard examples on which the next iteration of the boosting needs to concentrate.

The DataBoost algorithm, as shown in Figure 2, consists of the following three stages. During stage *A*, each example of the original training set is assigned an equal weight. The original training set is used to train the first classifier of the DataBoost ensembles. Secondly, during stage *B*, for each new classifier to be added to the DataBoost ensembles, a new training set is formed, which is used to train that new classifier. This new training set is formed by identifying hard examples based on both the weights of the training examples and the accuracy of the current trained classifier being added to the DataBoost ensembles. For each of these hard examples, a set of synthetic examples is generated. During the third Stage *C* of the algorithm, the synthetic examples are re-weighted and then added to the original training set to form the new training set. Finally, the weights of the new training data set are renormalized. Following the AdaBoost algorithm, the second and third stages of the DataBoost algorithm are re-executed until reaching the number of iterations as specified or the current component classifier's error rate is worse than a threshold value. Following the AdaBoost ensemble method, this threshold is set to 0.5 [4].

The detailed DataBoost algorithm is described next.



```
Algorithm DataBoost
Input: sequence of $m$ examples $\langle(x_1, y_1), \ldots, (x_m, y_m)\rangle$ with labels $y_i \in Y = \{1, \ldots, k\}$
       weak learning algorithm WeakLearn
       integer $T$ specifying number of iterations
Initialize $D_1(i) = 1/m$ for all $i$.
Do for $t = 1, 2, \ldots, T$

    1.1 Identify the hard examples from the original training data set
    1.2 Generate synthetic data based on the hard examples identified
    1.3 Re-weight the synthetic examples generated
    1.4 Add synthetic data to the original data set to form  a new training
        data set
    1.5 Call WeakLean, providing it with the new data set (Distribution Dt)

    2. Get back a hypothesis $h_t : X \rightarrow Y$.
    3. Calculate the error of $h_t$: $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$. If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop
    4. Set $\beta_t = \epsilon_t/(1 - \epsilon_t)$.
    5. Update distribution $D_t$: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & \text{otherwise} \end{cases}$
       where $Z_t$ is a normalization constant (chosen so that $D_{t+1}$ will be a distribution).

Output the final hypothesis: $h_{fin}(x) = \arg\max_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$.
```

Fig. 2. Pseudo-code of the DataBoost algorithm

3.1 Identify Hard Examples

During the first stage of the algorithm, the difficult to learn training examples are identified. Here, the DataBoost algorithm identifies a set of hard examples and uses them to generate synthetic data. The hard examples, namely the *seed* examples, are identified by the DataBoost algorithm as follows. Firstly, the number of hard examples N_s is calculated. N_s is calculated as the number of the original training examples multiplied by the error rate of the current trained classifier being added to the DataBoost ensemble. Secondly, the top N_s highest weighted examples are chosen as the seed examples, which will be used as seeds to generate synthetic examples. Note that the weight of each seed examples will be kept with the seed example and will be used for assigning proper weight to synthetic examples generated.

That is, after each trained classifier is added to the DataBoost ensembles, the weights of the 'easy' training examples being correctly classified by the current trained classifier are left unchanged. For the hard examples, the weights are increased by a multiplication factor. Thus 'easy' examples that are correctly classified by many of the previous trained classifiers get lower weights, and hard examples which tend often to be misclassified get higher weights. The weights of the examples reflect the examples' probabilities of being misclassified by the previous classifiers. Examples with higher weights are thus those which tend to be over-emphasized and need to be concentrated on by the DataBoost algorithm. Also, the accuracy of the current trained component classifier, which shows how many examples were misclassified by the current training, is used in the calculation of the number of seed examples.

3.2 Generate Synthetic Data

The data generation process generates a set of new examples that are based on each of the hard examples identified from the training set. These synthetic examples are added to the original training set to produce a new training set that is used to improve the classifier's current knowledge. Such new training set contains bias knowledge towards those particular examples and thus enhance the training process.

Even though the new training set is biased towards the hard examples, care is taken to ensure that the relevant knowledge as embedded in the original training set is maintained. This is addressed in three ways: Firstly, the new training set contains the original training set together with the newly generated training examples. Secondly, the data generation process is constrained by generating new data without changing the class distribution of the original training examples. Finally, the original distribution of attribute values in the original training set are used when constructing the new training set [12].

Figure 3 shows results of a hypothetical run of the data generation process. Assume that the large bold circle presents the hard example, the empty circle is the easy instance, and the small bold circle presents the synthetic instance. Figure 3 (a) shows a hard example that has been identified, depicted by the large bold circle. This hard example is subsequently used as the seed to generate synthetic data. Figure 3 (b) shows how a number of synthetic data which are related to the hard example have been generated, as indicated by the small bold circles. Also, Figure 3 (b) depicts the new training data set, which contains a number of additional synthetic examples. This combined data set will be used as training set for the component classifiers used in the next iteration of boosting.

In detail, the data generation process proceeds as follows [11, 12]. Consider a data set that contains n attributes $A_1 \ldots A_n$, where $n \geq 1$. The data set contains m classes $C \ldots C_m$, where $m \geq 1$. Each class C_k in the original training set has a total of E_k examples, where k=1…m. Assume that the hard example set contains a hard example R_j, which describes class C_k. It is the task of the data generation process to generate a new set of examples that reflects the knowledge contained in R_j. The class distribution of this set of examples reflects the original class distribution of $E_1 \ldots E_m$ examples.

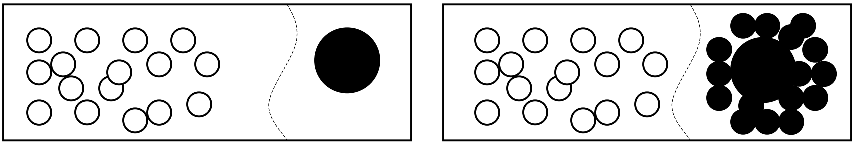

(a) Original training set **(b)** New training set

Fig. 3. Generating synthetic examples based on the hard examples to form new training data which contains synthetic examples with bias information and the original training examples

The first step of the data generation involves the generation of attribute values for each attribute A_i used in hard example R_j. Next, the data are formed into a new training set. The generation of values for each attribute A_i, as contained in R_j, is described next [11, 12]. Note that each of the attributes is considered separately.

- Suppose A_i denotes a **nominal** attribute. Consider R_j contains an attribute-value test ($A_i = x_i$), where x_i is an element of a finite set of possible values V. Here, E_k values describing class C_k that contains the value x_i for A_i are generated. For each complementary class C_q, the data generation process produces a total of E_q attribute values x_i, where $x_i \in V$ is randomly chosen to represent the distribution of values contained in the original training set with respect to class C_q.
- Assume A_i denotes a **continuous** attributes. Consider the attribute-value test ($A_i = x_i$). The data generation process proceeds to form, for each class C_k in the original data set a total of E_k attribute values that fall in the range [$minC_k \leq A_i \leq maxC_k$] are randomly generated. Here, $minC_k$ denotes the minimum value of attribute A_i with respect to class C_k and $maxC_k$ refers to the maximum value of A_i for class C_k, as found in the original training set. The mean value and variance of A_i with respect to class C_k is determined and the distribution of the values of A_i is constrained by generating values without changing the original mean value or the variance of the attribute.

The results of the first step are, for each attribute contained in example R_j , a total of $E_1 \ldots E_m$ attribute values that correspond to each of the classes $C_1 \ldots C_m$. The next step of the data generation process involves the merging of the attribute values into a set of training examples. This is achieved by, for each class C_k, combining the r^{th} value of attributes $A_1 \ldots A_n$, where $1 \leq r \leq k$, into one example [11, 12].

Interested readers are referred to [11, 12] for a detailed description of this process. Note that, in our previous work, as reported in [11, 12], data were generated based on high-quality *rules*, as extracted by the current base classifier. Here, high-quality rules were defined as those rules with accuracy higher than a pre-set threshold value. In contrast, in the DataBoost approach, the new examples are generated based on the *data itself*, i.e. on the seed examples as identified during Section 3.1. Also, each of the seeds is re-weighted before training the next classifier, as discussed next.

3.3 Re-weighting the Synthetic Data

Prior to the data generation process, a set of hard examples with weights that reflect the probabilities of being misclassified by the previous classifiers, are identified and used as seeds to generate new data. For each seed, the data generation process generates a set of synthetic examples. Each of synthetic examples will be assigned an initial weight corresponding to the weight of its seed used. The initial weight of each example is calculated by dividing the weight of the seed example by the number of instances generated from it. Assume that we have a seed example E_s with weight W_s, and a total of N_d examples have been generated based on the seed example E_s. For each example of the N_d synthetic examples, a weight W_s / N_d will be assigned by the DataBoost algorithm. In this way, the very high weights associated with the hard examples are balanced out. Note that, prior to training, the weights of the new training set will be renormalized, following the AdaBoost method, so that their sum equals 1[4,19].

4 Experiments

This section describes the results of evaluating the predictive accuracy of the DataBoost algorithm, in comparison with the AdaBoost method, using both decision trees and neural networks as base classifiers.

4.1 Data

To evaluate the performance of the AdaBoost and DataBoost methods, we obtained a number of data sets from the UCI data repository [15]. These data sets were carefully selected to ensure that they (a) are based on real-world problems, (b) are varied in characteristics, and (c) were deemed useful by previous researchers. Table 2 gives the characteristics of the data sets used for the experiments. Shown are the type of the features in the data set (i.e., nominal, continuous, or a mix of the two), the number of output classes, and the number of cases in the data set.

Table 2. Summary of the data sets used in this paper. Shown are the number of examples in the data set; the number of output classes; the number of continous and discrete input features.

Data set	Case	Class	Feature	
			Continuous	Discrete
Balance-scale	625	3	0	4
Breast-cancer-data	286	2	0	9
Contact-lenses	24	3	0	4
Congress-voting-1984	435	2	0	16
Hepatitis-domain	155	2	6	13
Horse-colic-data	368	2	7	15
Monk2	169	2	0	6
Monk3	122	2	0	6
Post-operative	90	3	0	8
Tic-tac-toe	958	2	0	9

4.2 Methodology and Experimental Results

We implemented the experiments using Weka [14], a Java-based knowledge learning and analysis environment developed at the University of Waikato in New Zealand.

Table 3. Test set accuracy rates for the data sets using (1) AdaBoost ensembles of decision trees; (2) DataBoost ensembles of decision trees (3) Improvement of DataBoost Approach (4) AdaBoost ensembles of Neural Network; (5) DataBoost ensembles of Neural Network; (6) Improvement of DataBoost Approach

Data set	C4.5			Neural Network		
	Ada Boost	Data Boost	Imp. over AdaBoost	Ada Boost	Data Boost	Imp. over AdaBoost
Balance-scale	77.92	76.16	-1.76	98.72	99.36	+0.64
Breast-cancer-data	65.38	67.83	+2.45	71.32	72.37	+1.05
Contact-lenses	66.66	83.33	+16.67	66.66	66.66	0
Congress-voting-1984	96.32	96.55	+0.23	95.17	95.17	0
Hepatitis-domain	81.29	84.38	+3.09	78.70	80.00	+1.3
Horse-colic-data	79.89	83.29	+3.40	78.53	80.97	+2.44
Monk2	69.67	74.55	+4.88	100.0	100.0	0
Monk3	92.59	93.51	+0.92	93.51	93.51	0
Post-operative	54.44	61.11	+6.67	56.66	61.11	+4.45
Tic-tac-toe	96.65	97.80	+1.15	96.97	96.97	0

For the Monk2 and Monk3 data sets [13], the entire population of 432 instances was used as the test data set, averaged over running the algorithm fifty times. Results for other data sets are averaged over five standard 10-fold cross validation experiments. For each 10-fold cross validation the data set is first partitioned into 10 equal sized sets, then each set is in turn used as the test set while the classifier trains on the other nine sets. For each fold an ensemble of 10 classifiers is created. Cross validation folds were performed independently for each algorithm.

For the decision tree we used the C4.5 trees as base classifiers [9]. For the neural networks we used standard back-propagation learning. The parameter settings for the neural networks include a learning rate of 0.3, a momentum term of 0.2, and the number of hidden layers is calculated by (attributes +classes)/2 [14].

Table 3 shows the test set accuracy rates for the data sets described in Table 2, for both the two decision tree and two neural network ensembles. We also show the results of the comparison of the DataBoost and AdaBoost algorithms. The table shows that the Databoost algorithm, in many cases, improved on the AdaBoost algorithm in terms of predictive accuracy. In particular, the DataBoost method produces equal or higher accuracies against all data sets, except a slight decrease of 1.76% against the balance-set data set when using decision tree component classifiers.

One conclusion drawn from the results is that the DataBoost method appears to consistently reduce the error rate for almost all of the data set, and in many cases this reduction is large. The DataBoost algorithm consistently produces good results across both neural network and decision tree ensembles. Furthermore, analysis of results suggests that the largest reductions in error are produced for small data sets. The post-operative data with 90 examples and the contact-lenses data with 24 instances obtained the largest error reductions of our experiments, notably 16.67% and 6.67% respectively when using decision tree component classifiers. Such largest reductions make sense, since for small data sets it is much easier to over-emphasize the hard examples. Moreover, the experiments demonstrated that the proposed approach can obtain larger performance gains while applying to domains where the learning accuracies are low. These results are promising, since they indicate that the DataBoost algorithm has application in domains where the number of examples is few.

5 Conclusions

This paper introduced a novel approach in which synthetic data are generated based on hard to learn examples as identified during training. The DataBoost algorithm was illustrated by means of a number of data sets from the UCI data set repository [15]. The results obtained indicated that the data generation approach increases the predictive power of boosting algorithms through augmenting the set of data used for training.

In conclusion, these results indicate two reasons for the accuracy improvement that is achieved by the DataBoost algorithm. The first is that the additional synthetic data with bias knowledge towards the hard examples prevent boosting from over-emphasizing the hard examples. The second effect is that, through generating synthetic data in the proposed solution, complementary knowledge about the hard examples is provided for the learning process to focus on.

It follows that our approach should be thoroughly tested against a large number of diverse data sets, using a variety of classifiers and ensemble schemes. Future work will thus include the application of this promising approach to both artificial and real-world data repositories. The use of other base classifiers and ensemble algorithms and other data generation methods will be further investigated. Also, the use of our technique with imbalanced data sets could lead to highly accurate results.

References

1. L. Breiman (1996) : Stacked regressions. Machine Learning, 24(1), 49-64
2. D. Wolpert (1992) : Stacked Generatlization, Neural Networks 5, 241-259
3. L Breiman (1996): Bagging Predictors. Machine learning, 24(2), 123-140
4. Y. Freund and R. Schapire (1996): Experiments with a new boosting algorithm. the Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, 148-156.
5. D. Opitz, R.Maclin (1999): Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence Research 11, 169-198
6. R. Shapiro, Y. Freud, P. Bartlett and W. Lee (1996): Boosting the margin: A new explanation of the effectiveness of the voting methods. Proc. Of 14th Intern. Conf. on Machine Learning, Morgan Kaufmann, 322-330
7. G. Ridgeway (1999) : the State of Boosting , Computing Science and Statistics, Vol. 31, 172-181
8. J. R. Quinlan: Bagging, Boosting, and C4.5 (1996) : In proceedings of the Thirteenth National Conference on Artificial Intelligence, 725-730. Portland, OR.
9. JR Quinlan, C4.5 (1994): Programs for Machine Learning, Morgan Kaufmann, California: USA.
10. 10.L.Breiman (1996): Bias, variance, and arcing classifiers. Tech. Rep. 460, UC-Berkeley, Berkeley, CA
11. HL Viktor and I Skrypnik (2001) : Improving the Competency of Ensembles of Classifiers through Data Generation, ICANNGA'2001, Prague: Czech Republic, April 21-25, 59-62.
12. HL Viktor (1999), The CILT multi-agent learning system, South African Computer Journal (SACJ), 24, 171-181.
13. SB Thrun et al (1991) : The Monk's problems: A Performance Comparison of Different Learning Algorithms. Technical Report CMU-CS-91-17. Computer Science Department, Carnegie Mellon University, Pittsburgh: USA
14. I. Witten, E.Frank (2000): Data Mining: Practical Machine Learning tools and Techniques with Java Implementations, Chapter 8, Morgan Kaufmann Publishers.
15. C.L. Blake and C. J. Merz (1998): UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA.
16. Y.Freund, R.E.Schapire (1999): A Short Introduction to Boosting, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780
17. T.G. Dietterich (2000): An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40, 139-157.
18. R.E. Schapire (1999): A brief Introduction to Boosting, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence
19. Y. Freund and R.E.Schapire (1997): A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.
20. H. Schwenk and Y. Bengio (1997): AdaBoosting Neural Networks: Application to On-line Character Recognition, International Conference on Artificial Neural Networks (ICANN'97), Springer-Verlag, 969-972.

Data Mining Approach for Analyzing Call Center Performance

Marcin Paprzycki, Ajith Abraham, Ruiyuan Guo, and Srinivas Mukkamala

Computer Science Department, Oklahoma State University, USA
Department of Computer Science New Mexico Tech
{marcin,aa,grui}@cs.okstate.edu, srinivas@cs.nmt.edu

Abstract. The aim of our research was to apply well-known data mining techniques (such as linear neural networks, multi-layered perceptrons, probabilistic neural networks, classification and regression trees, support vector machines and finally a hybrid decision tree – neural network approach) to the problem of predicting the quality of service in call centers; based on the performance data actually collected in a call center of a large insurance company. Our aim was two-fold. First, to compare the performance of models built using the above-mentioned techniques and, second, to analyze the characteristics of the input sensitivity in order to better understand the relationship between the performance evaluation process and the actual performance and in this way help improve the performance of call centers. In this paper we summarize our findings.

1 Introduction

The performance of the call center depends on the performance of its customer service representatives (CSRs) and the call handling regulations. Most existing large call centers collect data that is then used to assess and improve the performance of its representatives 12513. Typically, such data includes some form of quality assessment, time management, and business processing aspects 37. While, data mining has been applied to analyze the customer behavior with its main aim to improve the customer satisfaction, there is not much research on mining the data of performance of call center representatives. Therefore, the aim of our research is to fill this gap by applying data mining techniques to the combined performance evaluation results collected from five call centers of a large nationwide insurance company. The remaining parts of this paper are organized as follows. In section 2, we summarize the related research that was uncovered, followed by the short description of different data mining techniques used in our research (Section 3). Section 4 introduces the features of the data used in our study and introduces results of our experiments, including some sensitivity analysis. We briefly summarize our findings in Section 5.

2 Summary of Related Research

As indicated above, we were able to find only results related to mining customer-related data. Some vendors of monitoring system such as eTalk and GartnerGroup built data

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1092-1101, 2004.

mining tools into their *monitoring systems*. These tools are intended primarily for non-experts, such as supervisors and managers. They can "mine" the available data by asking "what if" questions 8. In this way it was found, for instance, that call transfers frustrate customers. *Predictive modeling* such as decision-tree or neural network based techniques can be used to predict customer behavior. Quaero LLC used such techniques to cluster customers according to their current and their potential value 10. *Textual data mining* has also been applied in the context of call centers. Busemann et al. classified e-mail request from customers based on shallow text processing and machine learning techniques. Their system was able to correctly respond to e-mails with an accuracy of 73% 11. Next, *audio data mining* has been experimented with. ScanSoft used context-free-grammar to parse the speech and follow by Sequence Package Analysis to caption the text to which data mining is applied. This approach allowed capturing early warning signs of caller frustration 4. Finally, *web usage mining* has been applied to web-based activities of call centers. Techniques utilized here are similar to these used in other cases of web mining 9.

3 Data Mining Techniques Used

Data mining is an information extraction activity with a goal of discovering hidden facts contained in data(bases). Using a combination of machine learning, statistical analysis, modeling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results. There exist a number of popular data mining techniques.

Multi-layer perceptron (MLP) is the most popular neural network architecture. It consists of at least three layers, an input layer of source neurons, at least one hidden layer of computational neurons, and an output layer of computational neuron(s). The input layer accepts inputs and redistributes to all the neurons of the middle layer. The neurons in the middle layer detect the features of input patterns and pass the features to the output layer. The output layer uses the features to determine the output patterns.

Linear neural networks (LNN) have just two layers: an input layer and an output layer. Linear models have good performance on linear problems. However, they cannot solve more complex problems. Linear networks can be trained to serve as a base comparison for non-linear problems. Linear model is relatively simple and not many parameters need to be selected by the users. We used the standard pseudo-inverse (SVD) linear optimization algorithm.

Probabilistic neural networks (PNN) have been developed for classification problems and utilize kernel-based estimation. They usually have three layers: one input layer, one hidden layer and one output layer. The network "embeds" the training cases into the hidden layer, which has as many neurons as there are training cases. The output layer "combines" the estimates and produces the output.

Classification and regression trees (CART) are techniques based on the tree structured binary decisions. Each decision tree has internal and leaf nodes. Leaf nodes represent the final decision or prediction. CART labels each leaf node a unique increasing integer number from left to right starting from 1. All the records in the dataset are assigned an integer. CART creates decision trees to predict categorical dependencies by using both categorical and continuous predictors.

Support vector machine (SVM) is a binary learning method 12. It conducts computational learning based on structural risk minimization that finds a hypothesis h for which the lowest true error is guaranteed. The true error of h is the probability that h will make an error for an unseen and randomly selected case. An upper bound of the true error can be used for h. Support vector machine finds the hypothesis h and minimizes the bound of the true error.

Finally, the above-described techniques can be combined and we have utilized a *hybrid* decision tree – neural network technique depicted in Figure 1. In this case, data is fed into the decision tree first and then the leave node information is obtained and added into the dataset used by the neural network as an additional variable (new attribute). For the neural network we have used the multi-layer perceptron with three layers and back-propagation learning for training. Here, the same training parameters were used as for the CART and the perceptron applied to separately to the problem.

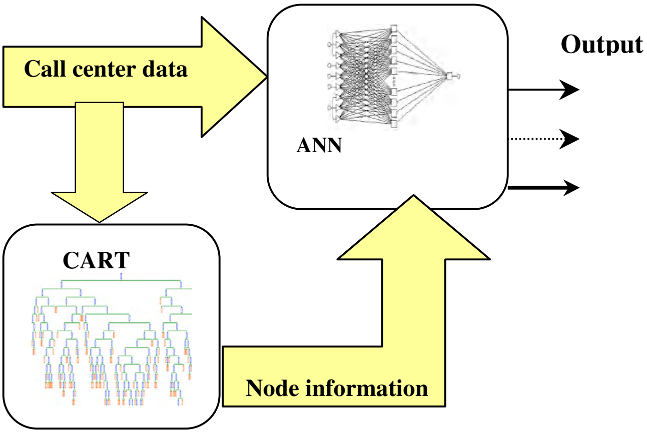


Fig. 1. Decision Tree-ANN Hybrid Model

4 Call Center Performance Data

The data used in this study is one year worth of actual data from the performance evaluation database of five call centers of a large nationwide insurance company. Here, each customer service representative is being evaluated monthly. To this effect randomly selected calls are recorded (out of ten to sixty calls answered daily by each representative) and the monitoring system constantly keeps up to ten calls for each CSR available. Of these, six randomly selected calls are used by a group of evaluators to assess the CSR's performance. In the insurance company from which the data was obtained, there are two main attributes against which the performance of its representatives is evaluated: (a) *customer service satisfaction* and (b) *business need satisfaction*.

The customer service satisfaction score is an *aggregate* result of evaluation based on eleven features. Exactly the same features are used for all products and all call centers.

Typical way of evaluating performance with regard to these features is by asking questions like: “did a CSR thanked the customer for calling the company?” or “did a CSR asked what else they can help customer with?” The result of the evaluation is an integer between 0 and 5. Here, 0 means that a given feature was not applicable to the call. A 1 indicates that the CSR did not meet the expectation. A 2 signifies that the expectation was met to some degree (denoted “met some”). A 3 indicates meeting the expectation. A 4 specifies exceeding the expectation. A 5 represents the case when the CSR far exceeded the expectation. These results are then aggregated to a value representing the total level of meeting the customer service satisfaction.

For example, an evaluator reviewed the call and found that only three questions out of eleven were applicable and marked them as 3, 4 and 1 according to how the CSR performed when she/he answered the call. The evaluator also marked the remaining eight questions as 0 (not applicable). The final score of customer service satisfaction was then calculated as the sum (8) divided by the number of applicable questions (3), resulting in the score equal to 2.67. The monthly score is the total score of all applicable questions of all six evaluated calls divided by the total number of applicable questions.

Business need satisfaction is scored exactly the same way as the customer service satisfaction. However, the features/questions vary from one product to another. Typical questions are “did a CSR provide correct information to customer” or “did a CSR access proper systems or documents.” Depending on the product, the minimum number of the questions is eight and the maximum is sixteen. Although the final scores of customer service satisfaction and business need satisfaction are continuous numbers ranging from 1 to 5, in the call centers, which were the source of the data used in the research, these results are converted to monthly evaluations according to the following rules:

Table 1. Rules for converting scores into final evaluation

not met	score < 2
met some	score >= 2 and score < 3
met	score >= 3 and score < 4
exceeded	score >= 4 and score < 4.75
far exceeded	score >= 4.75

Table 2. Dataset Description

Category	Attribute Name	Data Type	Format	Example
	Agent ID	Integer		1, 201, etc
	Date of Data	Date	mm/01/yyyy	09/01/2001
	Training	Boolean	0, 1	0
	Product ID	Integer		226, 3927
Quality	Customer Service	Category	1, 2, 3, 4	3
	Business Needs	Category	1, 2, 3, 4, 5	4
Time management	After Call Work Time	Integer	1, 2, 3,	180
	Adherence	Float	Percentage	96%
	Attendance	Integer	1, 2, 3, …	2
	Auxiliary	Float	Percentage	4%

In addition to the above, the attributes of *time management* are utilized and they are: *adherence*, *after call work time*, *auxiliary* and *attendance*. The data of time management is collected from phone switches on monthly basis. *Adherence* is the percentage of the length of time a CSR is logged into the phone switch to the length of time he/she is supposed to be logged in. *After call work time* is the average number of seconds that a CSR spends on post-processing data after calls during a given month. *Auxiliary* is the percentage of the length of time a CSR is spending on personal activity to the length of time that a CSR is logged into the phone switch. *Attendance* is a CSR's monthly absence.

Finally, in the available data, there is a Boolean attribute representing the fact that the CSR is / is not in a training period; each record has a time stamp; and there is an attribute representing which product a CSR is servicing. In summary, there are total ten attributes in the dataset utilized in our project and they are summarized in Table 2.

4.1 Data Cleaning and Preparation

As follows from the above, values of customer service and business need satisfaction should fall between one and five. We have therefore removed from the dataset all records with data outside of these bounds. The value of time management categories should all be equal to or above zero. The values below zero are not valid and were deleted. The records that had other missing values were also deleted from the dataset. Finally, when preparing the data, we have found that the distribution of the customer service satisfaction attribute was "bad." Only six records fell into the *not met* and thirteen into the *far exceeded* categories. These records were therefore deleted since they were too few to meaningfully participate in training and testing. The majority of the records fell into the *met* class. This class was thus separated into two sub-classes at 3.5. We have then utilized both the "big" met class and the "sub-class division" and compared the performance of models build for both cases. After cleaning, a total of 14671 records were left in the *customer service* dataset (1469, 5965, 5841, 1396 in subcategories, when the "met" class was separated) and 14690 records in the *business need* dataset (63, 3533, 5974, 3610, 1510 in each category).

Different products have different expected values of after call work, adherence and auxiliary categories. For example, 150 seconds may be a short after call work time for one of the products but a long time for another. Thus the after call work time, adherence and auxiliary were scaled to real numbers from the interval (0, 1). Finally, all of the remaining attributes, except date, were scaled similarly.

There are eight input attributes in the final dataset, which are agent ID, date, product ID, training, ACW, aux, adherence and attendance (see Table 2). There are two output attributes: customer service satisfaction and business needs satisfaction. To achieve the best performance, a separate model was built for each of the output attributes. There are four (three) possible output values for the customer service satisfaction and five values for the business needs satisfaction. All the algorithms use random sampling. Each experiment is repeated several times. The results from same algorithm very were close so we could make the assumption that the results are representative.

4.2 Experiment Setup

For the MLP we used one hidden layer. After a trial and error approach by varying the number of neurons from fifty to a hundred-twenty, we finalized the architecture with 113 neurons. There are eight neurons in the input layer since there are eight input attributes. There is one neuron in each model for one output. We used both a single-phase backpropagation based and a two-phase backpropagation (BP) combined with conjugate gradient (CG) training. We used a typical split of 50% data for training, 25% for testing, and the remaining 25% for cross validation. Same datasets were used for the different machine learning algorithms. We used 100 epochs for both back-propagation and conjugate gradient. In the PNN, we used 7337 neurons for training the customer service attribute and 7346 for business needs attribute in the hidden layer. In the CART algorithm, Gini was selected for goodness of fit measurement to achieve the best performance. We used a maximum tree height of 32 that resulted in the best performance. A hybrid decision tree-neural network was constructed as described in Section 3. For SVM's we used several kernels and after a trial and error approach, we used the third degree polynomial kernel, which resulted in its best performance.

4.3 Analysis of Predictive Performance

The performance measure is calculated from the classification accuracy of testing results. The performance result is the sum of total number correct prediction of the "correct" category and the correct prediction of the "wrong" category divided by the total number of testing cases. The performance of a perfect model is 100% for both the "correct" category and the "wrong" category. The models that have accuracy near 100% are "good." A random classifier should exhibit a 50% accuracy.

Table 3 shows the performance of each model for predicting customer service satisfaction. The results of the met class are shown in smaller font as a comparison of separated sub-classes. According to the overall results from the confusion matrix, the ranking of the performance of the trained models is CART, PNN, SVM, BP/CG, BP, Hybrid and the LNN. There are no apparent difference among the BP/CG, BP and the hybrid.

For example for the Met 1 class, there were 5969 records out of 14671 falling into "correct" category in the dataset and the remaining 8702 records fell into "wrong" category. CART predicted 4443 out of 5969 correctly, which was 74.43% shown as correct prediction of the "correct" class. CART predicted 6124 out of 8702 correctly, which was 70.37% shown in Table 5. Since 25% of the records in the dataset were used for cross validation for the LNN, MLP, PNN, and SVM, which is different from CART (10 fold cross-validation), the base to calculate the accuracy was different from CART, which was 3668. For example for the met 1 class again, 1448 records out of 3668 fell into the "correct" category and the remaining 2220 records fell into the "wrong" category. 873 records out of 1448 were predicted as "correct" correctly, which is 60.29%. 1359 out of 2220 were predicted as "wrong" correctly, which is 61.21% shown in Table 1. Table 1 also shows the accuracy details for customer service satisfaction. The research predicted the met class and also predicted each met sub-class by splitting the met class into two. Usually the prediction of one large class has higher accuracy. However, it is not true for the met class of customer service satisfaction. The performance for one large class is very close to the performance of predicting sub-classes indicating that the big class has more noise. Our research reveals that the scale used for customer service evaluation is incorrect and mixes data without good differentiation. The CSRs in sub-class 1 are more likely to be met-some performers. The CSRs in sub-class two are more likely to be exceeded performers.

Table 3. Classification Accuracy of Customer Service Prediction

Customer Service Skills – Cross Validation										
Class		**Case #**	**Linear** %	**BP** %	**CG** %	**BP/CG** %	**PNN** %	**CART** %	**Hybrid** %	**SVM** %
Met Some	Correct	1469	68.77	66.77	60.28	68.88	0.00	90.13	66.96	0.00
	Wrong	13202	66.67	70.71	58.91	70.68	100.0	83.08	70.47	100.0
	Overall		68.56	70.38	59.04	70.52	90.26	91.65	70.33	89.95
Met 1	Correct	5969	58.16	60.29	54.35	60.80	28.78	74.43	62.80	18.44
	Wrong	8702	60.31	61.24	54.77	60.73	86.37	70.37	58.66	90.64
	Overall		59.04	60.87	54.60	60.76	63.13	74.65	60.40	61.28
Met 2	Correct	5841	59.40	59.15	51.25	60.12	34.63	83.79	61.07	22.79
	Wrong	8830	59.93	61.75	52.85	62.88	81.55	63.59	61.95	88.65
	Overall		59.72	60.73	52.22	61.79	64.93	73.85	61.60	62.54
Met (1 and 2)	Correct	11810	55.77	61.29	47.46	60.87	99.79	74.69	61.88	100.0
	Wrong	2861	54.30	62.98	45.44	62.81	0.35	83.94	61.45	0.00
	Overall		55.49	61.62	47.07	61.25	89.57	76.50	61.61	80.30
Exceeded	Correct	1396	65.58	67.25	50.29	68.71	0.00	91.12	65.08	0.00
	Wrong	13275	63.32	68.51	49.14	68.72	100.0	84.12	71.43	100.00
	Overall		65.37	68.39	49.25	68.72	90.97	82.36	70.85	50.35

Table 4. Classification Accuracy of Business Need Prediction

Business need Satisfaction - Cross Validation										
Class		**Case #**	**Linear** %	**BP** %	**CG** %	**BP/CG** %	**PNN** %	**CART** %	**Hybrid** %	**SVM** %
Not met	Correct	63	50.00	53.85	53.85	53.85	0.00	100.00	65.00	0.00
	Wrong	14608	74.80	80.24	65.70	81.91	99.97	96.45	87.92	100.00
	Overall		74.73	80.15	65.66	81.81	99.46	99.62	87.80	99.73
Met some	Correct	3533	76.63	80.29	43.24	79.05	52.77	93.43	91.32	57.96
	Wrong	11138	75.33	81.14	40.66	81.90	91.73	83.38	82.63	90.59
	Overall		76.33	80.94	41.29	81.21	82.52	89.14	82.33	82.98
Met	Correct	5974	66.14	70.36	62.35	70.23	52.20	82.64	71.02	50.79
	Wrong	8697	60.30	68.03	59.38	67.94	81.40	75.03	69.07	90.59
	Overall		62.67	68.98	60.59	68.87	69.53	79.82	69.88	69.84
Exceeded	Correct	3610	68.31	73.77	55.77	74.22	23.10	93.82	76.52	24.57
	Wrong	11061	72.46	74.78	50.09	75.77	94.23	79.71	73.93	94.74
	Overall		71.46	74.54	51.53	75.38	77.12	86.51	74.59	76.75
Far exceeded	Correct	1510	71.03	74.92	59.22	75.83	2.12	96.82	78.00	0.00
	Wrong	13161	75.68	78.78	58.70	79.32	99.46	85.81	82.73	100.00
	Overall		75.17	78.43	58.74	79.00	90.69	92.33	80.84	96.12

Table 4 shows the performance of each model for predicting business need satisfaction. The way to calculate the performance of business need prediction is exactly the same as the way for customer service. The ranking of the performance is the same as the models for customer service. After looking into the performance accuracy of each correct/wrong class, the research found that PNN models are not valid for the dataset used. The performance of BP/CG is a bit better than BP. However the results are very close and it is not proper to make the conclusion that the models trained by BP/CG have better performance than the ones trained by BP alone. The performance of hybrid model was at least the same as CART. However, the overall accuracy is a bit better than BP and BP/CG models. The LNN model serves as a comparison for other models. The models trained by other algorithms are supposed to have at least the performance that linear models can get. CART models have the best performance in the research. They not only have the best overall performance, but also they have highest accuracy to predict "correct" (C1) and "wrong" (C0) for all each class.

Table 5. Ranking of the Inputs (importance) for Predicting Customer Service

Customer Service Satisfaction - Sensitivity Analysis										
Class	Algorithm	Agent	Date	Training	Product	ACW	Adherence	Aux	Attendance	Note
Met Some	Linear	7	1	5	3	4	2	8	6	
	BP	3	1	2	4	6	7	8	5	
	BP/CG	8	1	3	2	7	5	6	4	
	Hybrid	4	1	8	3	9	6	5	7	2
Met 1	Linear	3	1	5	4	6	8	2	7	
	BP	2	7	6	1	8	4	3	5	
	BP/CG	2	8	6	1	5	3	7	4	
	Hybrid	2	3	5	1	8	4	6	7	9
Met 2	Linear	2	1	5	7	8	3	4	6	
	BP	8	6	7	1	3	2	5	4	
	BP/CG	4	2	5	1	7	3	8	6	
	Hybrid	8	5	9	6	3	2	7	4	1
Met (1 & 2)	Linear	4	1	8	5	2	3	6	7	
	BP	8	1	3	7	6	5	4	2	
	BP/CG	8	1	3	7	6	5	4	2	
	Hybrid	7	1	4	6	3	5	8	9	2
Exceeded	Linear	2	1	7	3	5	6	8	4	
	BP	7	1	5	2	4	3	6	8	
	BP/CG	8	1	2	4	5	3	6	7	
	Hybrid	4	3	6	1	9	8	2	7	5

4.4 Inputs Sensitivity Analysis

The sensitivity is calculated by the accumulated errors when a particular attribute is removed from the training. When an attribute is removed from the training model, the higher the error is, the more important the attribute is. The importance of individual inputs is ranked by the accumulated error. Tables 5 and 6 illustrate the ranking of the various attributes for customer service and business needs prediction. First, product is very important to predicting customer service satisfaction, which indicates that CSRs in some products have more opportunity to far exceed than the CSRs in other products. Adherence is important too. Adherence is how much time of the required time a CSR spends logged into the switch and reveals the attitude toward work. A good attitude may lead to good customer service performance. Another interesting characteristic is that date is important when predicting customer service satisfaction. The reason why date is important may be that dates are interrelated with call types. One type of calls may be dominant of all types of calls during a certain period. After that period, calls of another type become the majority in the call volume in next period. Since we are not concerned about the call types in this research (no data is available to mine) we can only speculate that the affect of call types may materialize as the date parameter. Another way to explain the importance of the date may be the training or coaching delivery date. The customer service satisfaction may be improved right after the coaching or training session and may drop after a certain time afterwards. The ranking analysis from the LNN, BP, BP/CG and Hybrid model are pretty consistent in predicting business need satisfaction. The product becomes more important in predicting business needs satisfaction from not met class to the far-exceeded class. This can be interpreted that a CSR has more opportunity to be far exceeding if a CSR services a particular product and less opportunity if he/she services some other product. Agent is more important when predicting exceeded and far-exceeded classes. It means that the top performers are likely staying on the top most of the time. The performance of the CSRs whose performance falls into met or below met is not stable. However, they are more likely staying in met class or below.

Table 6. Ranking of the Inputs (importance) for Predicting Business Needs

Business Need Requirements - Sensitivity Analysis										
Class	Algorithms	Agent	Date	Training	Product	ACW	Adherence	Aux	Attendance	Note
Not Met	Linear	2	7	1	6	4	8	5	3	
	BP	7	1	2	6	4	8	3	5	
	BP/CG	7	1	6	8	4	5	2	3	
	Hybrid	4	5	9	2	8	7	3	6	1
Met Some	Linear	6	2	5	3	4	1	8	7	
	BP	4	3	7	1	8	2	5	6	
	BP/CG	4	3	5	1	8	2	6	7	
	Hybrid	4	2	3	1	8	9	5	6	7
Met	Linear	2	1	6	4	3	5	7	8	
	BP	7	1	4	2	8	3	5	6	
	BP/CG	8	1	6	2	7	3	5	4	
	Hybrid	7	3	6	2	8	4	5	9	1
Exceeded	Linear	6	8	4	3	2	1	5	7	
	BP	3	6	8	2	4	1	5	7	
	BP/CG	5	3	7	2	4	1	6	8	
	Hybrid	5	2	8	9	4	1	3	7	6
Far exceeded	Linear	2	3	7	1	6	5	4	8	
	BP	2	4	8	1	7	3	5	6	
	BP/CG	2	4	5	1	8	3	6	7	
	Hybrid	3	4	7	1	8	6	5	9	2

5 Conclusions

The research built six AI models to predict the quality score of customer service satisfaction and business need satisfaction by using LNN, MLP, PNN, CART, Decision tree-ANN Hybrid model and SVM. The research compared the performance of the six types of models based on the confusion matrix results of cross validation. The performance is also analyzed by using the accuracy of the "correct" category prediction and the accuracy of the "wrong" category prediction. The overall accuracy from CART is 80.63% on predicting customer service satisfaction and 89.48% on predicting business need satisfaction. The accuracy of the "correct" category and the accuracy of the "wrong" category are very close. The trained models based on CART can be used for future prediction. MLP training using BP and CG did not have significant better performance than BP alone. The research also analyzed the sensitivity of inputs. The research found that products, agents and dates could affect the quality of performance more than time management. The CSRs serving in some products have more opportunity to exceed the expectation than the ones in some other products. The top performers constantly exceed or far-exceed the expectation. The performance of CSRs whose evaluation results fall into met or below is not stable. The research suggest that call center management team should focus training and coaching the individuals and products that constantly have low quality instead of emphasizing balancing the length of times spent on calls.

References

1. A. Gilmore, Call Center Management: Is Service Quality a Priority, Managing Service Quality, vol. 11, no. 3 pp. 153-159.
2. A. Parasuraman, Service Quality and Productivity: a Synergistic Perspective, Managing Service Quality, vol. 12, no. 1, pp. 6-9.
3. Altitude Software, Improving Agent Performance While Maintaining High Level of Motivation, http://www.realmarket.com/required/altitude1.pdf, September 2002, access date: April 2003
4. Amy Neustein, Building Natural Language Intelligence into Voice-Based Applications, http://www.speechtechmag.com/issues/7_4/cover/915-1.html, July 2002, access date: April 2003.
5. David Sims, What Is CRM, http://www.crmguru.com, 2000, access date: April 2003
6. HigherGroud, Discovering the Business Intelligence Hidden in Your Call Center, HigherGroud White Paper Document, June 2002, access date: April 2003.
7. James Brewton, Customer Value-Driven Competency Models: Powerful Tools for Maximizing Call Center Performance and Customer Loyalty, http://www.crm2day.com, 2002, access date: April 2003.
8. Linda Dilauro, What's Next in Monitoring Technology? Data Mining Finds a Calling in Call Centers, Dictaphone Corporation.
9. Pan-Ning Tan, Vipin Kumar, Mining Indirect Associations in Web Data, WebKDD 2001: Mining Log Data Across All Customer Touch Points.
10. Ro King, Data Mining and CRM, http://www.crm2day.com, 2002, access date: April 2002
11. Stephan Busemann, Sven Schmeier, Roman G.Arens, Message Classification in the Call Center, Proceedings of the 6th Conference on Applied Natural Language Processing, Seattle, WA, 2000
12. Vladimir Vapnik, The Nature of Statistical Learning Theory. Springer, New York, 1995.
13. Warren Staples, John Dalrymple, Rhonda Bryar, Accessing Call Center Quality using the SERVQUAL Model, http://www.cmqr.rmit.edu.au/publications/wsjdrbicit02.pdf, 2002, access date: April 2003.

The Cognitive Controller: A Hybrid, Deliberative/Reactive Control Architecture for Autonomous Robots

Faisal Qureshi[1], Demetri Terzopoulos[1,2], and Ross Gillett[3]

[1] Department of Computer Science, University of Toronto,
Toronto, ON M5S-3G4, Canada
{faisal,dt}@cs.toronto.edu
[2] Courant Institute, New York University,
New York, NY 10003-6806, USA
[3] System Design Department, MD Robotics Limited,
Brampton, ON L6S-4J3, Canada
rgillett@mdrobotics.ca

Abstract. The Cognitive Controller (CoCo) is a new, three-tiered control architecture for autonomous agents that combines reactive and deliberative components. A behaviour-based reactive module ensures that the agent can gracefully handle the various real-time challenges of its environment, while a logic-based deliberative module endows the agent with the ability to "think ahead", performing more complex high-level tasks that require planning. A plan execution and monitoring module establishes an advisor-client relationship between the deliberative and reactive modules. We demonstrate CoCo in the context of space robotics—specifically the control of a vision-guided robotic manipulator that can autonomously capture a free-flying satellite in the presence of unforeseen mission anomalies.

1 Introduction

Humans are sophisticated autonomous agents that are able to function in complex environments through a combination of reactive behaviour and deliberative reasoning. Motivated by this observation, we propose a hybrid robotic control architecture, called the Cognitive Controller or CoCo, which combines a behaviour-based reactive component and a logic-based deliberative component. CoCo is useful in advanced robotic systems that require or can benefit from highly autonomous operation in unknown, time-varying surroundings, such as in space robotics and planetary exploration systems, where large distances and communication infrastructure limitations render human teleoperation exceedingly difficult. In our implementation, CoCo operates an autonomous, vision-guided robotic agent designed to service satellites in orbit. In realistic laboratory test scenarios, we subject CoCo to anomalous operational events, forcing its deliberative component to modify existing plans in order to achieve the mission goals. CoCo demonstrates the capacity to compensate in important ways for the absence of a human operator.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1102-1111, 2004.

In our space robotics application, we are specifically interested in the task of safely capturing a satellite, transporting it to a service bay, performing the desired service, and releasing it back into orbit. From the perspective of the software responsible for controlling the sensory apparatus and robotic manipulator, the first step is the most interesting and challenging. Once the satellite is secured, we can assume a static workspace and handle the remaining steps using more primitive scripted controllers [15].

Our autonomous robotic agent competently captures a satellite while handling anomalous situations such as sensor failures, hardware failures, and aberrant satellite behaviour. It gathers information about its environment through an imperfect vision system that is sensitive to lighting conditions and motion. CoCo determines whether the vision system is performing reliably, which is a non-trivial task that involves explaining current environmental events. If the explanation is unexpected, then either the vision system is failing or the environment is being erratic, and the agent must take corrective actions.

CoCo draws upon prior work in AI planning, plan-execution, mobile robotics, ethology, and artificial life. We review relevant prior work in the next section. In our technical development, Section 3 details the CoCo architecture and Section 4 describes its implementation. Section 5 presents results from CoCo's application. Section 6 concludes the paper.

2 Related Work

Early attempts at designing autonomous robotic agents employed a sense-model-plan-act (SMPA) architecture with limited success [9,12,13]. The 1980s saw the emergence of a radically different, ethological approach to robotic agent design, spearheaded by Brooks' subsumption architecture [4] and the mantra "the world is its own best model". Most notable among modern ethological robots is Sony Corporation's lovable robotic dog, AIBO [2], which illustrates both the strengths (operation in dynamic/unpredictable environments) and the weaknesses (inability to reason about goals) of the strict ethological approach. Hybrid architectures, containing both deliberative and reactive components, first appeared in the late 1980s. A key issue is how to interface the two layers. AuRA (Autonomous Robot Architecture) binds a set of reactive behaviours to a simple hierarchical planner that chooses the appropriate behaviours in a given situation [3]. In SSS (Servo Subsumption Symbolic), a symbolic planner controls a reactive module [6]. In ATLANTIS, the deliberative module advises the reactive behaviours [1,9].

2.1 Relationship to Previous Hybrid Architectures

Like ATLANTIS, CoCo consists of both deliberative and reactive modules, featuring a reactive module that performs competently on its own and a deliberative module that guides the reactive module. CoCo was originally inspired by experience implementing self-animating graphical characters for use in the entertainment industry. In particular, our approach was motivated by the "virtual merman" of Funge *et al.* [8], which extends a purely behavioural control substrate [14] with a logic-based deliberative layer employing the situation calculus and interval arithmetic in order to reason about discrete and continuous quantities and plan in highly dynamic environments.

CoCo differs in the following ways: First, its deliberative module can support multiple specialized planners where deliberative, goal-achieving behaviour is the result of cooperation between more than one planner. The ability to support multiple planners makes CoCo truly taskable. Second, CoCo features a powerful and non-intrusive scheme for combining deliberation and reactivity, which heeds advice from the deliberative module only when it is safe to do so. Here, the deliberative module advises the reactive module through a set of motivational variables. Third, the reactive module presents the deliberative module with a tractable, appropriately-abstracted interpretation of the real world. The reactive module constructs and maintains the abstracted world state in real-time using contextual and temporal information.

3 Cognitive Controller Architecture

CoCo is a three-tiered architecture that consists of deliberative, reactive, and plan execution and monitoring modules. The deliberative module implements a high-level symbolic planning system. The reactive module implements a low-level behaviour-based controller with supporting perception and memory subsystems (Fig 1). At the intermediate level, the plan execution and monitoring module enforces an advisor-client relationship between the deliberative and reactive modules.

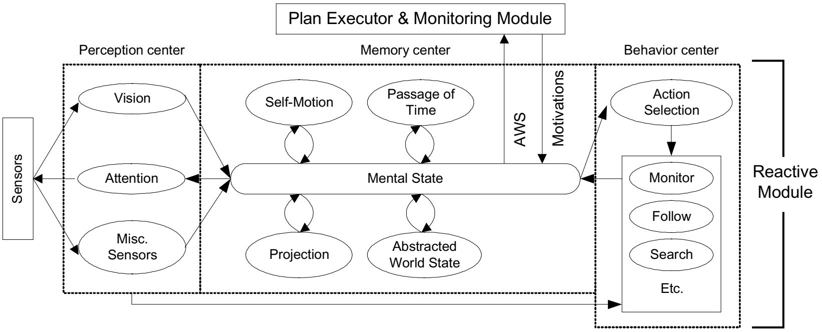


Fig. 1. The three functional components of the reactive module, each consisting of asynchronous processes.

3.1 The Reactive Module

CoCo's reactive module is a behaviour-based controller that is responsible for the immediate safety of the agent. As such, it functions competently on its own and runs at the highest priority. At each instant, the reactive module examines sensory information supplied by the perception system, as well as the motivational variables whose values are set by the deliberative module, and it selects an appropriate action. Its selection thus reflects both the current state of the world and the advice from the deliberative module. The second responsibility of the reactive module is to abstract a continuum of low-level details about the world and present a tractable discrete representation of reality within which the deliberative module can effectively formulate plans.

CoCo's reactive module comprises perception, memory, and behaviour components. This functional decomposition simplifies the design of the reactive module in order to implement basic behaviours, such as tracking, following, station-keeping, and capturing. These functional units are implemented as a set of asynchronous processes.

Perception Center. The perception center manages the vision system, which consists of long, medium, and short range vision modules. The long range module performs a search that returns an ongoing estimate of the satellite's pose once it has been detected. The estimate of the satellite's pose from the long range module initializes the medium range module, which is active from five meters to around two meters and uses model-based stereo-vision algorithms to track the satellite. The short range module takes over when the distance to the satellite is less than two meters. It tracks the satellite using visual features on the satellite's docking interface. The perception center decides which vision modules to activate and how to combine the information from these modules depending on their characteristics, such as processing times, operational ranges, and noise. An alpha-beta tracker filters out the noise from the vision readings. The perception center incorporates an attention mechanism that gathers information relevant to the current task, such as the status of the satellite chaser robot, the docking interface status, and the satellite's attitude.

Behaviour Center. The behaviour center manages the reactive module's behavioural repertoire. This by no means trivial task involves arbitration among behaviours. At each instant, the action selection mechanism chooses an appropriate high level behaviour by taking into account the current state of the world and the motivations provided by the deliberative module. We have implemented six such behaviours for the satellite servicing application—*search*, *monito*r, *approach*, *align*, *contact*, and *avoid*. The chosen action then activates lower level supporting behaviours, as necessary. The action selection mechanism chooses a behaviour that is relevant to the goals of the agent while ensuring its safety. The reactive module will heed the advice of the deliberative module only when it is safe to do so.

Memory Center. The memory center manages the short-term memory of the agent. It holds the relevant sensory information, motivations, state of the behaviour controller, and the abstracted world state. The robot observes its environment egocentrically. External objects change their position with respect to the robot as it moves. Behaviour self-motion constantly updates the internal world representation to reflect the current position, heading, and speed of the robot, otherwise the confidence in the accuracy of the world representation should decrease with time in the absence of new readings from the perception center (Fig. 2). The memory center filters out unnecessary details from the detailed sensory information consumed by the reactive module and it generates the abstracted world state (Fig. 3; Table 1) which expresses the world symbolically for use by the deliberative module.

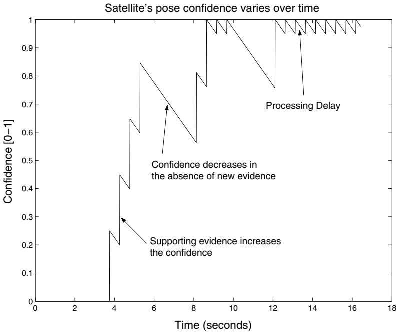


Fig. 2. Confidence in the satellite's pose decreases in the absence of sensory evidence from the vision system. How the confidence in a particular feature decreases depends on the feature (e.g., the confidence in the position of a dynamic object decreases more rapidly than that of a static object) and the penalty associated with acting on the wrong information.

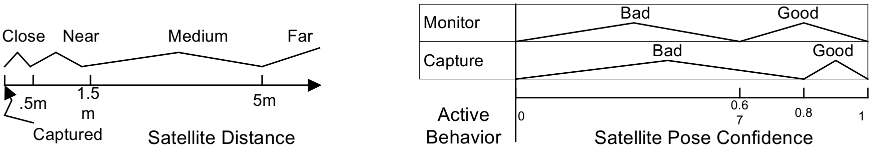


Fig. 3. The abstracted world state represents the world symbolically. For example, the satellite is either *Captured*, *Close*, *Near*, *Medium*, or *Far*. The conversion from numerical quantities in the memory center to the symbols in the abstracted world state takes into account the current state of the agent.

Table 1. The abstracted world state for the task of satellite servicing. The choice of fluents for describing the abstracted world state depends on the active task.

fStatus	fSatPosConf	fSatPos	fSatSpeed	fError
fLatch	fSatCenter	fSatAlign	fSensor	fSatContact

3.2 The Deliberative Module

The deliberative module endows our agent with planning ability, enabling it to perform high level tasks that are too difficult to perform without planning. The deliberative module maintains a set of planners, each with its own knowledge base, planning strategy, and task list. Each planner sees the world at an abstract level, which makes reasoning tractable, as opposed to planning among a myriad of low-level details. The reactive module determines the lowest level of abstraction for a planner, explicitly through the abstracted world state, and implicitly through its implemented behaviours (these behaviours—a.k.a., grounded actions—form the basis of the plans generated by the deliberative module). For any application, it is essential to choose the right level of abstraction in advance (Table 1).

Upon receiving a top-level command from the operator in the ground station, the deliberative module selects an appropriate planner (by using the task lists associated with the planners), updates the planner's world model using the abstracted world state, and activates the planner. Only one planner is active at a time in order to avoid unwanted interactions between actions proposed by different planners. The planner computes a plan as a sequence of zero (when the planner cannot generate a plan) or more actions, passes this plan to the deliberative module, which forwards it to the plan execution and monitoring module. Each action of an executable plan contains execution instructions, such as which behaviour to activate, and specifies its pre- and post-conditions.

Table 2. Grounded actions for the GOLOG planner - these actions are directly executable by the reactive module

aTurnon	aSearch	aGo	aLatch	aErrorHandle
aSensor	aMonitor	aAlign	aSatAttCtrl	aContact

Table 3. The deliberative module transforms the current task into a world state, called the desired world state. The Golog planner then constructs a plan to transform the current world state into the desired world state. Upon execution, this plan will fulfill the task.

fStatus(off) & fLatch(unarmed) & fSensor(medium,off) & fSensor(short,off) & fSatPos(medium) & fSatPosConf(no) & fSatCenter(no) & fAlign(no) & fSatSpeed(yes) & fSatAttCtrl(on) & fSatContact(no) & fError(no,X) **Initial (current) state**
fStatus(on) & fLatch(armed) & fSensor(medium,off) & fSensor(short,on) & fSatPos(zero) & fSatPosConf(yes) & fSatCenter(yes) & fAlign(yes) & fSatSpeed(yes)& fSatAttCtrl(off) & fSatContact(yes) & fError(no,X) **Goal (desired)**
aTurnon(on) -> aSensor(medium,on) -> aSearch(medium) -> aMonitor -> aGo(medium,near,vis) -> aSensor(short,on) -> aSensor(medium,off) -> aAlign -> aLatch(arm) -> aSatAttCtrl(off) -> aContact -> aGo(zero,park,no) **The plan that transforms the initial state into the goals state**

A Planner for the Satellite Capturing Task. Symbolic logic provides an appropriate level of abstraction for developing high level planners that elegantly express abstract ideas. We use GOLOG [11], an extension of the situation calculus, to develop a planner for the satellite capturing task. GOLOG uses logical statements to maintain an internal world state (fluents) and to describe what actions an agent can perform (primitive action predicates), when these actions are valid (precondition predicates), and how these actions affect the world (successor state predicates). GOLOG provides high level constructs, such as if-then-else and non-deterministic choice, to specify complex procedures that can model an agent and its environment. The logical foundations of GOLOG enable us to prove plan correctness properties, which is desirable.

The deliberative module updates the values of fluents from the abstracted world state and executes the GOLOG program. The execution generates a plan, such as the one in Table 3, whose purpose is to transform the current state of the world to the goal state—a state in which the chaser robot has securely captured the satellite. Depending on the script, this could be followed by more explicitly-scripted operations to service the satellite or it could continue to be directed through servicing operations by CoCo.

3.3 Plan Execution and Monitoring Module

The plan execution and monitoring module interfaces the deliberative and reactive modules. It initiates the planning activity in the deliberative module when the user has requested the agent to perform some task, when the current plan execution has failed or is otherwise unable to meet the desired post-conditions, when the reactive module is stuck, or when it encounters a non-grounded action that requires further elaboration. The execution is controlled through pre- and post-conditions specified by the plan's actions. Together, these conditions encode plan execution control knowledge. At each instant, active actions that have either met or failed their postconditions are deactivated. Next, un-executed actions whose preconditions are satisfied are activated (Fig. 5).

We divide actions into three categories: (i) grounded actions (directly executed by the reactive module; see Table 2), (ii) conditional actions (affect the choice of the next action to be executed), and (iii) non-grounded actions (require further elaboration, such as the user command *dock*). The plan executor and monitoring module can handle all three categories of actions, so it can execute linear, conditional, and hierarchical plans. It can also execute multiple actions, and hence multiple plans, simultaneously; however, it assumes that the plan execution control knowledge for these plans within the deliberative module will have prevented race conditions, deadlocks, and any undesirable side affects of concurrent execution.

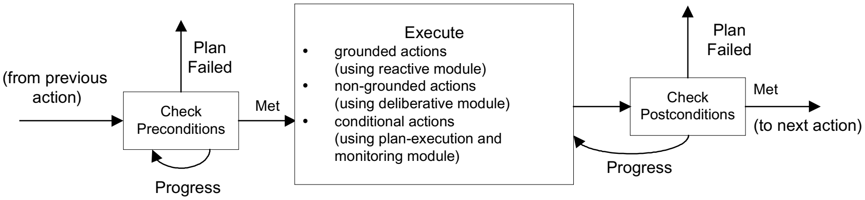
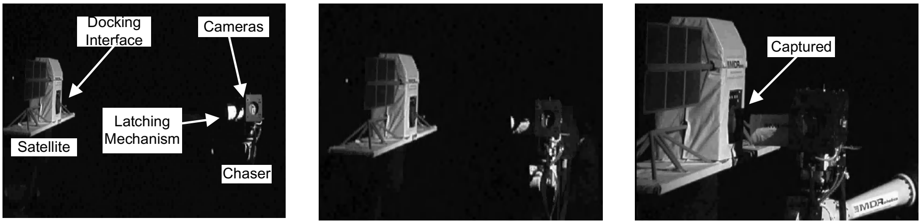

Plan Failed
(from previous action)
Check Preconditions
Progress
Met
Execute
• grounded actions (using reactive module)
• non-grounded actions (using deliberative module)
• conditional actions (using plan-execution and monitoring module)
Plan Failed
Check Postconditions
Met
(to next action)
Progress

Fig. 5. The plan executor and monitoring module sequentially executes each action. It checks the current action preconditions until they succeed or fail, checking preconditions and postconditions before each transition.

4 Implementation

To facilitate the design, development, and debugging of CoCo, we implemented an Autonomous Agent Design and Simulation Testbed (AaDST)—a software testbed for developing autonomous agents in virtual environments. In our application, AaDST contains a physics-based model of the chaser robot, a kinematically controlled satellite that can exhibit realistic satellite motion by following prescribed trajectories, and a virtual sun whose position affects lighting conditions. The virtual chaser has synthetic visual sensors that model the characteristics of the actual vision system, including processing delays, noise characteristics, and lighting effects. The virtual chaser also provides motor commands that are similar to those of the physical robot.

The physical setup consisted of MDRobotics Ltd. proprietary "Reuseable Space Vehicle Payload Handling Simulator", comprising two Fanuc robotic manipulators and their associated control software (Fig. 6). One robot with the stereo camera pair mounted on its end effector acts as the chaser. The other robot carries a grapple fixture-equipped satellite mockup and generates realistic satellite motion. The robot lab is specially designed with black walls, ceiling and floor to mimic the lighting conditions of the space environment—very little ambient light, strong sunlight and harsh shadows.



Chaser searching for the satellite

Chaser approaching the satellite

Chaser successfully captured the satellite

Fig. 6. The chaser robot captures the satellite using vision in simulated orbital lighting conditions.

First, we implemented the controller for the virtual chaser—it takes input from the synthetic vision system and issues motor commands to the virtual chaser. Next, we modified it to control the physical robot by adding the necessary communication modules that enable it to obtain sensory data from the actual vision system and issue motor commands to the physical robot over the local intranet. As we had hoped, this transition from controlling a virtual chaser in a simulated environment to controlling the physical robot required only minimal changes to CoCo. We merely had to modify the low-level behaviours in the reactive module, because the dynamic responses of the physical robot were not identical to those of the virtual chaser.

5 Results

Typically, the ground station would upload a mission plan to the on-orbit sequencer that would then pass to CoCo the relevant commands (Fig. 7). However, as a benefit of the level of autonomy that CoCo affords this system, only a single high-level command, *dock(time-out_in_seconds, max_attempts)*, needs to be uploaded from the ground station in order to perform the free-flyer satellite capture operation in which we are interested. We tested CoCo in the simulated environment and also on the physical robots, and it met its requirements; i.e., safely capturing the satellite while handling numerous anomalous situations. We performed 800 test runs in the simulated environment and over 25 test runs on the physical robots. CoCo never jeopardized the safety of the satellite or the chaser. For each run, we randomly created error conditions (see Table 4), such as vision system failure and hardware failure. CoCo's chaser robot gracefully handled all of them, successfully capturing the satellite whenever it was able to recover from these failures. In situations where it could not resolve the error, it safely parked the manipulator and informed the ground station of its failure.

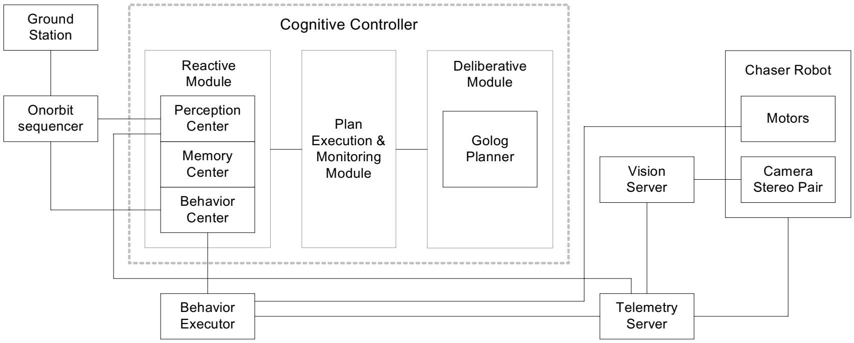


Fig. 7. CoCo receives commands from the ground station through the on-orbit sequencer and controls the chaser robot.

Table 4. CoCo handled these error conditions that were randomly generated during various test runs.

Vision System Errors	Hardware Errors
Camera failure Self shadowing Solar glare Failed transitions between vision module	Grapple fixture errors Joint errors (critical) Satellite's attitude control error

6 Conclusion

Toward the design of intelligent, hybrid controllers for autonomous agents, CoCo advocates general principles that address the critical challenge of combining reactivity and deliberation for autonomous robots inhabiting complex, dynamic environments. CoCo features a behaviour-based reactive system and a logic-based deliberative system, and it provides an elegant scheme for combining the two through a plan execution and monitoring system. Safety is ensured by a fully competent reactive module that can override when necessary the suggestions of the deliberative module. The CoCo architecture is also taskable, because its deliberative layer allows multiple planners specialized to the various tasks the agent must perform. Our domain of application, space robotics, requires an autonomous robot to deal with a myriad of operational events and anomalies that require real-time response. Within this domain, we have successfully implemented a controller that meets these challenges. Another benefit of our approach is that the uploaded command set from the operator is greatly simplified due to the ability of the on-orbit system to decide among the various operational details autonomously. The proposed architecture will apparently be useful in developing intelligent hybrid controllers for autonomous agents in other domains.

Acknowledgements. The authors thank MD Robotics Limited and Precarn Associates for funding this work, and acknowledge the valuable technical contributions of Dr. P. Jasiobedzki, H.K. Ng, S. Greene, J. Richmond, Dr. M. Greenspan, M. Liu, and A. Chan.

References

1. P. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of American Association of Artificial Intelligence*, San Mateo, CA, 1987.
2. R. C. Arkin, M. Fujita, and R. Takagi T., Hasegawa. Ethological modeling and architecture for an entertainment robot. *ICRA*, 2001.
3. R. C. Arkin. Integrating behavioural, perceptual and world knowledge in reactive navigation. *Journal of Robotics and Autonomous Systems* (**1-2**), June 1990, 6, 1990.
4. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, **RA-2**, 1986.
5. W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz,W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of AAAI/IAAI*, 1998.
6. J. Connell. SSS: A hybrid architecture applied to robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992.
7. R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, **5**(2), 1971.
8. J. Funge, X. Tu, and D. Terzopoulos. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proceedings of the Conference on Computer Graphics (Siggraph99)*, 1999. ACM Press.
9. E. Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mo- bile robots. In *Proceedings of the National Conference on Artificial Intelligence*, 1992.
10. D. Hdhnel, W. Burgard, and G. Lakemeyer. GOLEX—Bridging the gap between logic (GOLOG) and a real robot. In *Proc. of the 22nd German Conf. on AI (KI-98)*, 1998.
11. Y. Lesp´erance, R. Reiter, F. Lin, and R. Scherl. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, **31**, 1997.
12. E.D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, **5**(2), 1974.
13. E.D. Sacerdoti. The nonlinear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, 1975.
14. D. Terzopoulos, X. Tu, and R.Grzeszczuk. Artificial fishes: Physics, locomotion, perception, behaviour. *Artificial Life*, **1**(4), 1994.
15. R. Gillett, M. Greenspan, L. Hartman, E. Dupuis, D. Terzopoulos. Remote Operation with Supervised Autonomy (ROSA). In *Proceedings of the 6th International Conference on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2001),* 2001.

Intelligent Systems Integration for Data Acquisition and Modeling of Coastal Ecosystems

Carl Steidley, Alexey Sadovski, and Ray Bachnak

Department of Computing and Mathematical Sciences
Texas A&M University Corpus Christi
Corpus Christi, TX 78412 USA
{steidley,Sadovski,rbachnak}@falcon.tamucc.edu

Abstract. In this paper, we describe the primary data acquisition system for environmental modeling efforts at Texas A&M University-Corpus Christi. Additionally, we describe the design and development issues encountered in the production of two supplemental data acquisition systems. Finally, we present examples of the use of the acquired data for water level prediction modeling efforts.

Keywords: Intelligent Systems, Intelligent Interfaces, Neural Networks.

1 Introduction

One of the research goals of the Department of Computing and Mathematical Sciences (CAMS) is to develop effective and reliable tools for modeling the environmental systems of the Gulf of Mexico. Our modeling approaches are based on the real-time data collected by the Texas Coastal Ocean Observation Network (TCOON). TCOON is managed by the Division of Nearshore Research (DNR) at Texas A&M University-Corpus Christi (A&M-CC) in cooperation with CAMS. TCOON consists of approximately 50 data gathering stations located along the Texas Gulf coast from the Louisiana to Mexico borders.

Water quality data collection for our modeling efforts in the very shallow water areas, less than 3 feet, of these bays and estuaries is a challenging task. Obstacles encountered in such environments include difficulty in covering large territories and the presence of inaccessible areas due to a variety of reasons such as a soft bottom or contamination. There is also a high probablity of disturbing the test area while placing the sensors. To address these issues, a number of research centers have been developing autonomous boats [3-5]. These boats, however, require course planning prior to deployment. As a result, the course is not easily changed once the boat is in the water. To supplement and expand TCOON, we have designed and developed a remotely controlled shallow draft vehicle (boat) that continuously and efficiently collects water quality in shallow water areas (6 in-3 ft), rather than using fixed position sensors.

Advances in imaging technology and sensors have made airborne remote sensing systems viable for many environmental data acquisition applications that require

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1112-1122, 2004.

reasonably good resolution at low cost. Airborne images provide a higher temporal resolution with superior spatial resolution as sources of information for various modeling applications including vegetation detection and environmental coastal science analysis. Remotely sensed images, for example, can be used to study the aftermath of and model such episodic events as hurricanes and floods that occur year round along the south Texas coast. To supplement our TCOON data acquisition capabilities further, we have designed and developed an airborne multi-spectral imaging system (AMIS), which uses digital cameras to provide high resolution at very high rates. System software is based on Delphi 5.0 and IC Imaging Control's ActiveX controls. Both time and the GPS coordinates are recorded.

2 The Texas Coastal Ocean Observation Network (TCOON)

In 1989 the Division of Nearshore Research (DNR) at A&M-CC commenced the installation of a modern state-of-the-art water-level system along the Texas coast. The first measurement systems installed by DNR were intended to provide real-time water-level and meteorological information to the City of Corpus Christi to assist local officials with preparations for incoming hurricanes and tropical storms. From this initial work, other state agencies such as the Texas General Land Office and the Texas Water Development Board began contracting DNR to provide similar information for other areas along the Texas coast. Following a Texas Legislative mandate in 1991, this network of water level gauges became the Texas Coastal Ocean Observation Network (TCOON). As a result, TCOON expanded from an initial three stations in Corpus Christi in 1989 to over forty stations by 1992 [1].

TCOON is presently operated in cooperation with the Texas General Land Office, the Texas Water Development Board, the U.S. Army Corps of Engineers, and the National Oceanic and Atmospheric Administration (NOAA). The network utilizes the Next Generation Water Level Measurement System developed by NOAA's National Ocean Service (NOS) for the National Water Level Observation Network. Water-level data and other environmental parameters are made available in near-real time via packet radio, cellular telephone, and the Geostationary Operational Environmental Satellite (GOES). The TCOON system has been installed using the NOAA criteria for guidance.

Today, the TCOON system has more than 50 stations, including seven long-term stations established and operated by NOS as part of its National Water Level Observation Network. NOS guidance has been followed to obtain reliable water level data and to address legal concerns for the admissibility of the data in the determination of littoral boundaries. Most stations provide additional data such as wind speed and direction, air temperature, and water temperature, and some stations provide water current, salinity, pH, and dissolved oxygen data. Data collected as part of TCOON are available via the Internet at *http://dnr.cbi.tamucc.edu/*.

2.1 TCOON Station Configuration

Each TCOON station has sensors for measuring the various environmental parameters, a data collection computer for controlling the sensors and temporarily storing on-site data, one or more telemetry devices for retrieving data from the station, and solar panels and batteries for power. Most TCOON stations use a Next Generation Water Level Measurement System (NGWLMS) as designed by NOS [2]. At the heart of the NGWLMS system is a Sutron 9000 computer that controls the attached sensors, collects and stores environmental observations, and transmits the observations via satellite, radio, or telephone modem. The Sutron 9000 has been quite reliable for TCOON operations; however these systems are beginning to deteriorate with age and it is difficult to find repair or replacement parts. As a result, in the past several years DNR has gradually migrated to using Vitel VX1100 Data Acquisition and Telemetry Unit computers for data collection at some stations. The VX1100 system provides much of the same functionality and capability as the Sutron 9000 at a lower cost.

For other projects such as the Corpus Christi Real-Time Navigation System and the Freeport FlowInfo System, DNR has developed a data collection computer using industry-standard PC-104 computer components. A typical PC-104-based data collection computer consists of an Intel 486 or Pentium™ processor, 16 megabytes of RAM, and at least 20 megabytes of solid-state hard disk space. The data collection computer runs a modified form of the Linux operating system; this allows the use of a rich set of software development tools and provides a robust multitasking environment for controlling sensors, storing data, and doing on-site processing such as data compression for transmission over bandwidth-limited communications channels.

All the stations in TCOON and related projects measure environmental parameters at some multiple of six-minute intervals. For example, water-level measurements are taken every six minutes, while other measurements such as salinity or barometric pressure may be made every thirty or sixty minutes. The data are stored in on-site memory by the data collection computer and then transmitted to DNR by one or more communications channels including satellite, spread-spectrum packet radio, or telephone modem. At stations where radio or telephone connections are not available, satellite transmissions are used to transmit data at hourly or three-hourly intervals. Thus, the time from measurement to acquisition at CBI depends on the measurement interval and the communications medium used. The data arrive at DNR somewhere between six minutes and six hours after measurement.

3 A Shallow Draft, Remote Controlled Vehicle

Data collection in shallow water areas normally requires setting up sensors in several places. In addition to being redundant and time consuming, this task when performed manually has a high chance of disturbing the test area. Currently, in areas not covered by TCOON stations we collect water quality data in areas with water 3 feet or deeper by a man-controlled boat. In this section we describe a project undertaken by an interdisciplinary team of CAMS scientists along with environmental investigators at DNR to design and develop a remotely controlled shallow draft boat that continuously and efficiently collects water quality in shallow water areas (6 in. to 3 ft.). Our boat is small in size (7 ft. in length and 3 ft. in width), has a shallow draft, and can be easily steered to collect data in real-time. The prototype is designed to collect salinity and other environmental data and is equipped with onboard computers, water quality instruments (Hydrolab), GPS, digital compass, a remote control receiver and a receiver/transmitter radio (Freewave). It also has sensors to detect objects from all directions (front, sides, back, and bottom) so as to intelligently maneuver around obstacles. Acquired data is stored and transmitted wirelessly via a radio to a remote control station in real-time and data is logged to a PC for later processing.

3.1 System Design

Operational requirements considered in designing the boat included the following: (a) Remote control within the operator's line of sight, (b) Small size and an ability to transport without extra towing equipment, (c) Stability to resist waves and wind, (d) A draft as small as 6 inches, (e) Ability to detect objects in all directions, and (f) Wireless transmission of acquired data in real-time.

Embedded System PC and Sensors. The onboard PC consists of a stack of PC/104 modules, called the "Cube," with analog-to-digital conversion capabilities and serial port interfaces. The cube acts as a central control unit and interfaces with the radio and all onboard sensors, including the GPS and digital compass. The water quality sensor is a Hydrolab® designed to be used in fresh, salt, or polluted water. This instrument measures several parameters, including temperature, pH, dissolved O_2, and salinity. Our Hydrolab® model utilizes a pump via a tube to take the water through the process onboard, making it very useful in the shallow water areas in which we use our system since the Hydrolab® does not have to be immersed in water [6].

3.2 Testing the Prototype

Our first "sea test" was performed primarily to determine that the boat draft met the design goal of a six-inch or less draft. We also wanted to gather experimental data to determine the optimal locations of the compartments where the waterproof case and the battery was to be permanently placed. The test was completed on December 17, 2002. The draft was measured at two different places: 1) the bow of the boat and 2) the transom of the boat. The test was conducted first without any load and again with all components expected to be present during an operation (trolling motor, marine battery, and waterproof case filled with the electronic components used for propulsion control and data collection). The following table summarizes the results.

Table 1. Boat draft test results

Boat condition	Draft at bow (in)	Draft at transom (in)
Empty	1	1.5
Loaded	2	3

The test revealed some major accomplishments: the boat met the draft design-specification and remained stable in rough water conditions with and without the load.

4 Airborne Multi-spectral Imaging System

Airborne remote sensing has many applications that include vegetation detection, oceanography, marine biology, and environmental coastal science analysis. The integration of remote sensing and geographic information systems (GIS) in environmental applications has become increasingly common in recent years. Airborne, multi-spectral images of earth's surface are excellent sources for scientific information. There are many multi-spectral Satellite Remote Sensors such as the LANDSAT MSS and LANDSAT TM, but these systems offer only 30-meter spatial resolution pixels. Another limitation of satellite sensors is that their temporal resolution is based on their orbital passes.

A&M-CC is located on Ward Island in Corpus Christi Bay. Hurricanes, tropical storms and other episodic events occur year round in the area. Our marine biologist, environmental scientist, and geologist colleagues of the Center for Coastal Studies often need to investigate and analyze the after effects of such events at near real time. The vendors of satellite images are often unable to meet the temporal and spatial resolution needed by our colleagues. In addition to being costly, commercial airborne remote sensing systems have to be deployed at the scene of investigation as soon as possible for data to be useful for studies and research. This section of this paper describes an Airborne Multi-Spectral Imaging System (AMIS) and presents test results that clearly demonstrate the capabilities of the system [7 - 9].

4.1 Prototype System

The prototype consisted of one Sony DCR-PC1 MiniDV handycam. Our configuration, like the airborne remote sensing system at Ohio State University or the digital camera system at the University of Calgary, required data to be recorded and post-processed. Although this solution delays the availability of results, it produced good spatial results, for example, 2-meter positional accuracy and 3-meter accuracy in height [10]. Recently, a small-format aerial photographic system was used in combination with lower resolution images for rectification [11]. Compared to existing scanned products, the digital frame array offers a pixel resolution of around 4.5 μm.

The system was mounted in the inspection hole of a Cessna 172RG (Courtesy Lanmon Aerial Photography Inc.) The camera was suspended within the hole by means of a mounting ring and arm attachment. We used aluminum alloy as the material for the mount ring. Under the mount ring a rubber ring was placed to dampen the vibration of the camera. To prevent wind from entering the aircraft around the sides of the video camera, a hood was made to fit over the lens housing. The hood extended from the lens of the camera to the skin of the aircraft and was constructed from foam-rubber material. This foam-rubber hood served two purposes; it eliminated the wind and exhaust from entering the aircraft around the video camera, as well as, further reduced the vibration of the mounted system.

A test flight was completed on March 1, 2002 by taking aerial pictures of the A&M-CC campus and the immediate coastal area. A number of ground control points were set on campus. A LCD screen (an 8 mm Sony Digital Player) mounted on the pilot's control allowed the pilot to see what the video camera was viewing. The LCD screen and the video camera were connected through an S-video cable. The images were recorded on both the digital videotape and by utilizing Pinnacle Studio DV version 7 software on a laptop with an Adaptec Inc. IEEE 1394 PCMCIA (firewire) interface card. Results showed that the use of high-resolution digital cameras meets the needs of the scientific staff at A&M-CC. The aerial images of the prototype system have a 20 cm resolution for a flying height 1500 feet.

4.2 Global Positioning System and Software

After the initial test flight, the prototype was upgraded to include a Garmin GPS III Plus receiver used to record the latitude, longitude, and altitude (from the NMEA-0183 protocol string) of the acquired images. Delphi was selected for programming the system. Delphi is a high-level, compiled language that supports structured and object-oriented design. Its benefits include easy to read code, quick compilation, and use of multiple files for modular programming. Delphi was used to control the camera and the GPS. The software enables the user to display and record the video to the system hard disk. The GPS receiver was connected to the COM port of the system computer. The software reads and records the GPS co-ordinates and the corresponding time. The software incorporates ActiveX controls (The Imaging Source GmBh), which enables easy access to imaging devices connected to the computer. ActiveX Controls from Imaging Source GmBh was used to play the frames in a video window in the software GUI. The user can record the video into the hard disk drive. IEEE-1394 (firewire) interface provides speeds up to 400 Mbps and can easily transfer digital video. The GPS co-ordinates and the video are synchronized by recording the start time and end time of the video. This file provides the frame and the exact GPS co-ordinates for a given time. Synchronization of the video and GPS co-ordinates is done using flat-file approach.

4.3 Testing Upgraded System

In early July 2002, there was massive flooding in south Texas. On 12th July 2002, a second test flight was conducted over the Chapman Ranch located in Nueces,
Kleberg, and King counties of south Texas. The purpose of the flight was to determine the quality of the system's vegetation detection as well as record the flooding of the Nueces River. A three-band digital CCD sensor was used to capture the extents of the flood area. For this application it was determined that the three bands would allow for the determination of the spectral signature for determination of the limits of the flood.

For vegetation detection, we flew over the Chapman ranch at lower altitude (3000 ft.) before capturing the Nueces River flooding at 12,500 feet. Some of the images were processed and analyzed. The images were also enhanced to improve convolution, edge enhancement, and contrast utilizing a spectral pass filter. The image was rectified using existing Digital Ortho Quarter Quadrangle images from the US Geological Survey. Next, the images were enhanced to improve convolution, edge enhancement, and contrast utilizing a spectral pass filter The next objective was to replace visual colors with a classified pattern. An unsupervised classification method was used to determine the natural breaks between the shapes, sizes and spectral signature. The objective of the development of the classified map is to (i) identify a flooded area and its boundaries, and (ii) assign landownership to flooded area. A more involved method of reclassification was used to identify land cover types. A six-color classification was performed using spectral pattern recognition of the Jenks natural breaks. This image groups similar spectral signature items for classification. Land cover is represented by the natural and artificial compositions covering the earth surface and are used to assess the flood impact.

A new digital camera, Sony DFWSX900, has been acquired to improve the resolution and speed of the system. The final multi-camera system will have multi-spectral capabilities with the GPS receiver providing position and height of the images. Further, an inclinometer sensor will be incorporated to provide accurate heading and tilt information.

5 Water Level Statistical Modeling

Tide charts, based on harmonic analysis, are generally the method of choice for forecasting water levels. Tide charts are mostly based on astronomical forcing or the influence on water levels of the respective motions of the earth, the moon, and the sun. There are locations around the world, including the Gulf of Mexico, where other factors such meteorological forcing often dominate tidal forcing [12] and limit significantly the application of tide charts. In such cases other models must be developed to accurately forecast water levels. We have considered three different models for "next-hour" predictions, and two of these produced quite reliable predictions. The first of these models is a multi-regression model in which the "next-hour" prediction is based on the levels of water, speeds and directions of wind for the previous 48 hours with a step of 2 hours. This model did not produce the expected results. The coefficient of correlation for these predictions was less than 0.5. The second approach was another multi-regression model in which two-hour predictions of water level are based on the levels of water during the previous 48 hours, using 2-hour steps. Here we now believe that information about weather (pressure, wind, temperature, etc.), used in the model previously described, is hidden in the levels of water (See Factor Analysis below.). This model worked remarkably well: R squared for all stations was greater than 0.95. To make further predictions we used the previously determined levels of water. Such a step by step approach produced quite good predictions. The third approach was also based on linear multi-regression of the levels of water, first differences, and second differences for such levels for the previous 48 hours with the step equal to two hours.

This approach produces the same quality of water level prediction as the second approach, i.e. $R^2 > 0.95$. These results are quite understandable, since in both cases we have to deal with linear combinations of previous water levels. The difference in these two models is as follows: the third approach has between four (4) and eight (8) significant variables in a linear regression while in the second model of linear regression we use all twenty four (24) variables where these variables are the water levels for the previous 48 hours.

After analyzing different regression models we faced the following question, "Why do models with only previous water levels work much better than models with all kinds of meteorological data provided by TCOON stations?" To answer this question we applied factor analysis to the water levels over the period of 48 hours with an interval of 2 hours. The conclusion is that no more than 5 factors explain over 90% of variance for water levels for all TCOON stations. Then we compared the results of factor analysis for shallow waters with results of factor analysis for deep water stations. We have discovered the following:

- In coastal shallow waters and estuaries the major or the first component is not periodical, and we call this component "weather". Other main components are periodical and we call them "astronomical".
- In off-shore deep waters, the first two or three components are astronomical components, while weather is a less dominant component.
- Our conclusion is that the prime factor affecting water levels in estuaries and shallow waters is weather.
- It has been observed also, that linear regression models for different locations have different coefficients for the same variables. We think that this difference may be explained by the geography of the place where the data is collected.

These conclusions assisted us in improving predictions in the shallow waters since the conclusion suggested integrating regression approach with harmonic analysis. Namely, we use the idea that variations of water levels depend on two things – harmonic component (which is called tides) and another component which is affected by weather. Let us denote

$$x_n = w_n - h_n \tag{1}$$

where: x_n is the difference between water level wn and harmonic water level h_n at the moment n

Then we can apply a technique, which is similar to that used for our statistical model described above. That is, we can predict next hour difference between water level and harmonic level

$$x_1 = a_0 x_0 + a_{-1} x_{-1} + \ldots + a_{-n} x_{-n} \tag{2}$$

and step by step

$$x_k = a_0 x_{k-1} + a_{-1} x_{k-2} + \ldots + a_{-n} x_{k-n} \tag{3}$$

Now we can find prediction for the water levels as follows

$$pw_t = h_t + x_t \tag{4}$$

This approach to predictions of water levels proved to be very effective.

6 ANN Modeling and Predictions

The Artificial Neural Network (ANN) modeling approach is also based in forecasting future water level differences based on past water level differences. Other inputs to the ANN model have also been tested. For example, past wind squared is included in the model discussed below; as it has been recognized that wind forcing is well correlated with water anomalies. Other inputs, such as barometric pressure, have been tested but models which included past water level differences, past wind measurements and wind forecasts have been shown to be optimal [12]. It has also been shown that simple neural networks with one hidden layer and one output layer have the best performance [12], [13]. With one input neuron with a tansig function and one output neuron with a purelin function and a number of total different inputs ranging from 10 to 30 the ANN forecast of a water level n hours beyond the time of forecast can be expressed as follows:

$$x(t_o + n) = a + \left(\frac{2b}{\left(1 + e^{-\left(c + \Sigma d_i y_i \right)} \right)} \right) - 1$$

In the expression above, the additive parameters (a, c) are identified as the model biases and the multiplicative parameters (b, d_i) are referred to as the model weights. These parameters of the ANN are defined in the process of training of neural network over the known set of data. The y_i are the input to the model. The exponential terms in the ANN model provide a non-linear modeling capability.

The training of ANN models is different in nature as compared to the methods for our statistical model. There is typically no demonstrated method to identify a global optimum. The goal of the training process is therefore to find a suitable local optimum. To identify a good local optimum ANNs are trained over past data sets starting with a random guess of the model parameters and using the repeated comparison between the output of an ANN and an associated set of target vectors to optimize the weights of the neurons and biases of the model. All the ANNs discussed in this work were trained using the Levenberg-Marquardt back-propagation algorithm and implemented using version 4.0 of the Matlab Neural Network Toolbox and the MATLAB 6.0 Release 12 computational environment running on a Pentium PC.

The performance of the ANN for the prediction of water levels was tested at the Bob Hall Pier, Texas, TCOON station. The model was trained and tested using three data sets composed of 3600 hourly measurements of water levels, wind speeds and wind directions. The data sets covered the spring seasons of 1998, 2000, and 2001 from Julian day 21 to Julian day 182. The model was successively trained on each data set and applied to the other two data sets. This procedure provided a set of six time series of predicted water levels to be used for validation. For each time series the average absolute error between predicted and measured water levels was computed. Averages and standard deviations were then computed for the results of the six validation time series for these two parameters. The standard deviation gives an overall measure of the variability due to the differences between training sets as well as the differences resulting from the training process. The inputs to the model were selected as the previous12 hourly water level and wind measurements based on experience gathered during the modeling for other locations [13]. One model was trained without wind predictions while for the second case wind measurements were used to simulate wind forecasts. These wind forecasts consisted of future wind measurements at 3 hour intervals up to 36 hours. A database of wind forecasts is presently being constructed and models based on wind forecasts are expected to be more representative of future model performance. The ANN model captures a large fraction of the water anomaly and improves significantly on the tide tables. The performance of the models with and without wind forecasts is compared with the performance of the tide tables for forecasting times ranging from 6 to 36 hours. Both ANN models improve significantly on the tide tables for forecasting times up to 24 hours. Improvements for 30-hours and 36-hours predictions are still measurable. The addition of wind forecasts improves the model performance although not significantly as compared to the improvement over the tide tables.

References

1. Michaud, P.R., Thurlow, C, and Jeffress, G.A, “Collection and Dissemination of Marine Information from the Texas Coastal Ocean Observation Network,” *U.S. Hydrographic Conference Proceedings, Hydrographic Society Special Publication No. 32*, 168-173, Norfolk, Virginia, 1994.
2. Mero, T.N. and Stoney, W.M., “A Description of the National Ocean Service Next Generation Water Level Measurement System”, *Proceedings of the Third Biennial NOS International Hydrographic Conference*, Baltimore, Maryland, April 1988.
3. B. Ross, “A Robot Boat for Offshore Science,” The Robotics Institute-Carnegie Mellon University, United State, 2002.
4. R. Rocca, “Launching the Roboat,” Circuit Cellar Journal, Vol. 115, pp. 32-38, February 2000.
5. Woods Hole Oceanographic Institution, “Remote Environmental Monitoring Units,” http://adcp.whoi.edu/REMUS/index.html, September 2002.
6. R. Kulkarni, R. Bachnak, S. Lyle, and C. Steidley, “Design and development of an airborne multi-spectral imaging system,” Proceedings of the 2002 AeroSense Conference, April 1-5, 2002, Orlando, FL, Vol. 4725, pp. 540-546.
7. R. Bachnak, R. Kulkarni, S. Lyle, and C. Steidley, “Airborne Multi-Camera System for Geo-Spatial Applications,” Proceedings of the 2003 AeroSense Conference, Vol. 5097, Orlando, FL., 2002.
8. M.M. Mostafa, K.P.Schwarz, and P. Gong, “GPS/INS Integrated Navigation System in Support of Digital Image Georeferencing,” Proceedings of the Institute of Navigation 54th Annual Meeting, Denver Colorado, pp. 435-444, 1998.
9. M.M. Mostafa, and K.P. Schwarz, “A Multi-Sensor System for Airborne Image Capture and Georeference,” *Photogrammetric Engineering & Remote Sensing*, 66 (12), pp. 1417-1423, 2000.
10. Sadovski, A. L., P. Tissot, P. Michaud, C. Steidley, Statistical and Neural Network Modeling and Predictions of Tides in the Shallow Waters of the Gulf of Mexico. In "WSEAS Transactions on Systems", Issue 2, vol. 2, WSEAS Press, pp.301-307.
11. Tissot P.E., Cox D.T., Michaud P., Neural Network Forecasting of Storm Surges along the Gulf of Mexico. *Proceedings of the Fourth International Symposium on Ocean Wave Measurement and Analysis (Waves `01)*, ASCE, 1535-1544, 2002.

12. P.E. Tissot, D.T. Cox, and P.R. Michaud, “Optimization and Performance of a Neural Network Model Forecasting Water Levels for the Corpus Christi, Texas, Estuary”, 3rd Conference on the Applications of Artificial Intelligence to Environmental Science, Long Beach, California, February 2003.
13. Sadovski, A. L., C. Steidley, P. Tissot, P. Michaud, "Developing a Goodness Criteria for Tide Predictions Based on Fuzzy Preference Ranking", Developments in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence 2718, , pp. 391-401, Springer, June 2003.

Extracting Patterns in Music for Composition via Markov Chains

Karsten Verbeurgt, Michael Dinolfo, and Mikhail Fayer

Department of Computer Science
State University of New York at New Paltz
75 S. Manheim Blvd., Suite 6
New Paltz, NY, 12561
Verbeurg@cs.newpaltz.edu

Abstract. The goal of this paper is to describe a new approach to algorithmic music composition that uses pattern extraction techniques to find patterns in a set of existing musical sequences, and then to use these patterns to compose music via a Markov chain. The transition probabilities of the Markov chain are learned from the musical sequences from which the patterns were extracted. These transitions determine which of the extracted patterns can follow other patterns. Our pattern matching phase considers three dimensions: time, pitch, and duration. Patterns of notes are considered to be equivalent under shifts in time, the baseline note of the pattern, and multiplicative changes of duration across all notes in the pattern. We give experimental results using classical music as training sequences to show the viability of our method in composing novel musical sequences.

Keywords: Intelligent Systems; Data Mining; Machine Learning

1 Introduction

The prospect of a computer composing music autonomously is enticing. It addresses the more fundamental question of whether machines can achieve creativity in a manner comparable to that which we accept as artistic creativity in humans. It is perhaps even more enticing because music is a form of art that pervades our daily lives, and has done so since the origins of civilization. Music is a medium that is particularly well suited to such study, in contrast to drawing or painting, because we can capture music in a score that can be formalized into a machine representation. In this paper, we give a new technique for autonomously composing music that learns from established examples of musical sequences. This work can thus be viewed as cognitive modeling, in that we construct a program that attempts a task that is usually thought to require human creativity.

On a more technical and practical level, the music composition problem is an assembly problem. In our approach, we study sequences of actions, find common subsequences, learn what subsequences can follow each other, and recompose them in novel ways. This is a problem that generalizes to other areas. Consider the problem of dressing in the morning. The components of dress are well established: socks,

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1123-1132, 2004.

shoes, shirts, pants, and jackets, which come in different colors (or tones). There are certain subsequences of the dressing problem that we do the same way day after day. Being human, however, we typically vary the color in different ways. We may change the order of some subsequences, such as shirt before pants, or pants before shirt, while some other sub-sequences can only occur before others, such as socks before shoes or shirt before jacket. The goal of our approach applied to this problem domain would be to study sequences of dressing over periods of time, and learn to come up with new and different combinations that still achieve the implicit rules.

Music composition is an inherently difficult endeavor even for the most skilled human experts. One technique a composer may utilize is to incorporate patterns, or *motives,* from other pieces. Our approach is analogous to this method of composition, in that we extract patterns from previously composed music, and use them to compose new music. We extract patterns from existing music by using a suffix tree data structure. Our pattern detection technique considers three dimensions: time; pitch; and duration. Patterns of notes are considered to be equivalent under shifts in time, the *baseline pitch* of the pattern, and multiplicative changes of duration across all notes in the pattern (the *baseline duration*). After extracting a set of patterns, our goal is to compose new music. The challenge we face in composing new music given patterns is to sequence them in a way that preserves musical soundness.

The technique we employ for sequencing patterns is to construct a Markov chain whose states represent the patterns extracted from the training pieces, and whose transition probabilities are learned from these sequences. At each state of the model, we must also choose a baseline pitch and duration to transform the pattern in those dimensions. Selection of baselines is accomplished via probabilistic emission functions at each state in the model, whose probabilities distributions are also learned from the training sequences. We do not provide any music theory rules to the algorithm. We are thus implicitly learning the music theory rules that guide composition using patterns, or motives. In contrast, most algorithmic composition techniques do provide music theory knowledge to the composition algorithm [2,3,4].

2 Related Work

There are two separate bodies of literature related to this work. One is the area of algorithmic music composition. Many researchers have investigated autonomous music composition over the decades, with the first attempts dating back to the early origins of the electronic computer. In 1956, Hiller and Isaacson produced the first composition by computer, "The Illiac Suite for String Quartet". We will not attempt to survey the vast literature on algorithmic composition here, but instead refer the reader to the book of Miranda [8] for such a survey. Here we only mention that previous work on algorithmic composition has utilized neural networks, evolutionary computation, formal grammars, automata, Markov chains, and fractals [2,3,4,8,12]. The approach we propose in this paper is most similar to prior work on Markov chains for music composition. While previous work has used states of the Markov model to represent single notes, our work differs in that the states of the Markov chain may represent patterns of notes rather than single notes. The music composed thus has the potential for a much richer structure.

The other body of literature related to this work is that of pattern extraction in music. Much less work has been done in this area than in the area of algorithmic composition, and the work that has been done is relatively recent. The focus of this work has been on retrieval of existing music from large databases. The primary application of pattern detection in music retrieval is to allow the user to search for a piece of music based upon a segment of the music that is known. This approach is taken in the Greenstone project of at Waikato University [1,13]. Another approach to musical information retrieval has been developed at MIT [5]. The goal of their system is to allow the user to whistle a small segment of a piece of music, and to have the system retrieve the entire song.

Other work by Lemstrom and Laine [6] has proposed using the suffix tree data structure to index musical sequences. The pattern detection part of our work is closest to their approach, since we also use the suffix tree data structure to extract patterns from existing music. Our goal, however, is to use the patterns extracted from many pieces of music as a repertoire to be used for composition of new pieces, rather than to retrieve music based upon queries.

The idea of using a database of patterns to compose music is not new. It was used by well-known composers, at least as far back as Mozart. The work of David Cope [3] uses earlier Mozart symphonies to try to compose pieces similar to later symphonies. One can view our approach in a similar manner. We are essentially learning background knowledge to aid in the composition process. We now proceed to a description of the pattern extraction technique we use, the suffix tree.

3 Suffix Trees for Pattern Extraction

The first step in our method of learning to compose new music from existing pieces is to extract patterns from the pieces. Our longer term goal of this project is to construct a large pattern database from which more and more diverse musical arrangements could be constructed. To demonstrate the method, however, for the purposes of this paper we will be using only a few short training pieces.

The method we use for pattern extraction is a data structure called a suffix tree [7,11]. This data structure has been in the literature since the 1970's, but has experienced more interest in the last decade due to its applications in indexing web documents and DNA sequences. We will not give a detailed exposition of suffix trees in this paper, since this can be readily obtained elsewhere. Rather, we give a brief description of the operations of the suffix tree for pattern extractions, for the reader who is not familiar with this data structure.

In order to understand the suffix tree data structure, it is perhaps easiest to first describe a related structure called a *suffix trie*. A *trie* is a multi-way tree, with one branch possible for each character in the alphabet of the document that is being indexed (see Fig. 1). In the case of musical sequences, we can think of the trie as containing one branch for each of the twelve possible notes. A sequence of notes is then stored in the trie as a path in the multi-way tree. A suffix trie is a trie that indexes every suffix of a sequence, starting at each position in the sequence. Each subsequent note in the sequence is inserted into the trie until the suffix starting at that position is distinguished from every other suffix. More precisely, for every position i, from 1 to the length of the sequence, we insert the subsequence starting at position i into the trie, until we reach a leaf. At that location, we store the index i. Since every leaf in the trie has only one index associated with it, the sequence, or pattern, on the path to that leaf is unique in the original sequence (i.e., it occurs only once.) However, all of the internal nodes in the trie correspond to patterns that occur more than once in the sequence.

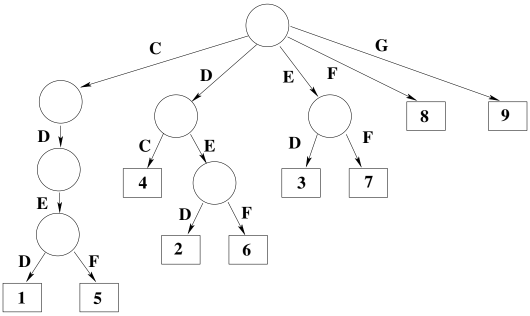


Fig. 1. This suffix trie is constructed for the sequence of notes C D E D C D E F G. Starting at each position in the sequence, we insert the subsequent letters of the sequence until we have a unique suffix. Note from the trie that the pattern C D E occurs twice in the sequence, once starting at position 1, and once at position 5. The pattern starting at position 1 is distinguished from that starting at position 5 by the fourth letter, which is a D for the occurrence at 1, and an F for the occurrence at 5.

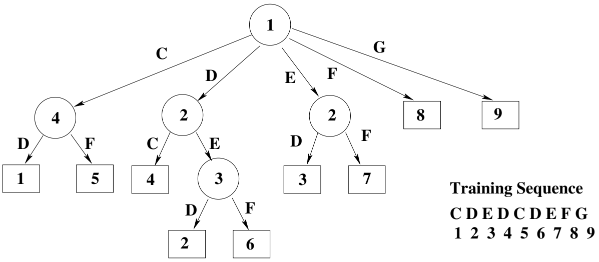


Fig. 2. The suffix tree in this figure corresponds to the suffix trie in Fig. 1. Note that the C D E path in Fig. 1 that contains no branching has been compressed into a single node.

Now, a *suffix tree* is a compressed form of the suffix trie. One can think of the suffix tree as a suffix trie in which each of the internal paths that has no branching is compressed into a single node. Each node contains the position number *in the pattern* for the character at that node. This indicates the number of nodes that have been compressed (see Fig. 2). The characters of the compressed nodes can be retrieved from the original text, since each leaf node contains the index of the occurrence of that pattern in the original sequence

We can now state more precisely what we consider to be a pattern in a musical sequence. A *pattern* or *motif* in a musical sequence is a right-maximal subsequence that occurs j times in the sequence for some integer j, where right-maximal implies that if the sequence is extended on the right by one note, then the resulting subsequence will occur strictly less than j times in the whole sequence. This notion of pattern corresponds to what is represented by the nodes of the suffix tree (internal nodes correspond to patterns that occur more than once, and leaves correspond to patterns that occur exactly once.)

3.1 3-D Pattern Recognition via Baselines

For clarity of exposition, in the previous section we considered only patterns in the time dimension. That is to say, if the sequence C D E occurs at two different points in the music, then it would be considered as a pattern. We also consider patterns that reoccur in the music at a different pitch and with different durations. Thus, the sequence G A B would be considered as the same pattern as C D E. The first pattern is said to occur at a *baseline* of G, and the second at a baseline of C.

The representation we use for patterns in the suffix tree is thus based upon the differences, or intervals, between successive notes rather than on the actual note values. The sequence C D E would therefore be represented as +2 +2 with a baseline of C, and the sequence G A B would be represented as the same sequence of offsets, +2 +2, but with a baseline of G. It is these sequences of offsets between notes that are stored in our suffix tree. We allow for offsets of 128 to –128 to represent the entire piano keyboard.

We also consider the durations of notes in our pattern matching. A sequence that matches in the dimensions of time and pitch, but that differs by a multiplicative constant in the duration of the notes is considered to be a match. In our system, we allow whole, half, quarter, and sixteenth notes. The multiplicative constants for duration can therefore be 16, 8, 4, 2, 1, ½, ¼, 1/8, and 1/16. We refer to this quantity as the duration baseline.

By using baselines for pitch and duration, we are performing three-dimensional pattern recognition, with dimensions of time, pitch, and duration. We argue that this is a more meaningful notion of pattern in music, since sequences such as C D E and G A B, described in the previous two paragraphs, sound like the same pattern to the listener, even if duration is changed from quarter notes in the first sequence to eighth notes in the second sequence. Similar notions of baselines for pattern matching are employed in two-dimensional image compression techniques such as J-peg.

Once the patterns in a set of training pieces have been extracted, we then use these patterns in the composition of new pieces. The method we use for composition is a Markov chain, which is described in subsequent sections.

4 Composition Using a Markov Chain

A Markov chain consists of a finite number of states, a probabilistic transition function that indicates the probability of making a transition from one state to another, and an initial state. Each state in the model may also have an output, or *emission*. We refer the reader who is unfamiliar with Markov chains to [9,10].

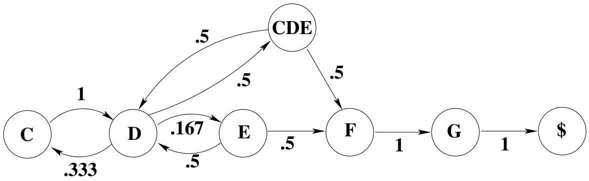


Fig. 3. The Markov chain we construct for music composition comprises a set of nodes representing individual notes, and a set of nodes representing more complex patterns from the internal nodes of the suffix tree. The node CDE represents the pattern C D E that occurred twice in the training sequence of Fig. 2. In our implementation, we only include pattern nodes that are of length three or greater. Transition probabilities are determined by the number of times each pattern follows another in the training sequences, weighted by the length of the pattern to give a preference for using longer patterns.

In previous work that uses Markov chains for music composition [8], there is one state for each note of the scale, and the transition probabilities determine the probability that a note of one value follows a note of another value. The ``creativity" in music composed with such a model is due to the randomness of the transitions. This randomness comes at a price, however, since it also places limitations on the amount of structure that can be achieved in the composition. This use of Markov chains is only capable of modeling rudimentary music theory concepts. For example, transitions between notes such as C and D in the key of C can be given a relatively high probability, while transitions from C to F would be lower, and transitions from C to C# would be small. This allows a modeling of the preference for smooth continuous transitions of pitch within the key, but allows also a small probability of producing accidental notes. While this use of Markov models does allow for a rudimentary probabilistic encoding of basic music theory rules, it is extremely limited in the structure that it can introduce into the music, since it only considers music one note at a time.

The approach we take in this work is a more complex use of Markov chains to compose music. We augment the basic Markov chain discussed in the previous paragraphs with states that represent the internal nodes in the suffix tree (see Fig. 3). We then learn the transition probabilities from the example music that was used to construct the suffix tree, as is described in the next section.

4.1 Learning the Transition Probabilities

We learn the transition probabilities for the Markov chain from the example training sequences that were used to form the set of patterns in the suffix tree. Recall that each internal node of the suffix tree represents a pattern in the training sequences that occurred at least twice. For each such occurrence of a pattern, we determine which patterns from the suffix tree (i.e., which internal nodes) can follow it. We then assign transition probabilities to each of these "next" patterns in proportion to the length of the next pattern. That is to say, a transition to a next pattern of length 2 will receive a relative weight of 2, and a transition to a next pattern of length 3 will receive a relative weight of 3. Values are then normalized to be probabilities (see Fig. 3). The effect of this weighting is to give a preference for using longer patterns from the suffix tree, but still allowing the use of shorter (less structured) patterns. More formally, if a pattern P1 is followed three different patterns P2, P3 and P4 in the training sequences, with P2 having length one, P3 having length 2, and P4 having length 3, then the transition probabilities would be 1/6, 1/3, and ½, respectively.

4.2 Baseline Emissions

As discussed previously, our pattern recognition technique allows for patterns to differ in their baseline pitch or in their baseline duration and still be considered the same pattern. Thus, the node CDE in the model of Fig. 3 can also represent G A B, and any other baseline offset. In addition, the notes C D E could be quarter notes, while the G A B notes are eighth notes. The pattern node thus actually represents the sequence of intervals starting at the baseline note, that is to say, a baseline of C followed by +2 half steps, then another +2 half steps. When arriving in a state of the model, therefore, we need to generate a baseline note for pitch as well as duration. This is done via a probabilistic emission function, with the probability of each baseline for a pattern being determined by the frequency of that baseline for that pattern in the training sequences.

We have now described the states of the Markov model, representing patterns of notes, and the transitions of the model, representing the probabilities of one pattern following another. To complete the model, we now need to describe the set of initial states, and how to terminate the musical sequence.

4.3 Initial and Final States

A Markov chain must have an initial state, or a set of initial states and a probability distribution on the set. We use the later technique of multiple initial states. For the initial states, we select the set of states that begin the training sequences. In the results described in this paper, the distribution over initial states is uniform. Thus, the generated sequences must begin with the same pattern as one of the training sequences. This might at first glance seem restrictive if there are only a few training pieces, as is the case in the results presented in this paper. Since each pattern may be transformed in pitch and duration, however, it allows for many possible variations of the starting notes. We chose these initial states because we wanted to distinguish opening sequences from other patterns in the pieces, due to the fact that music typically has an opening sequence that might not occur later in the piece, and with characteristics that may be particular to opening sequences.

Since we are using a Markov model whose transitions from each state are typically non-zero, the model could potentially run infinitely, producing an infinitely long musical sequence. Since most listeners' attention spans are finite, we need to design a method of terminating the generated sequences. There are two natural approaches to ending the musical sequence generated by the Markov model: specifying terminating states; or terminating the sequence after a fixed number of transitions. The former of these methods would be analogous to our method of determining initial states. That is to say, and sequence that terminates a training piece can also terminate the generated sequence. For the results of this paper, however, we chose the latter method, and terminate the sequence after a fixed number of transitions (we chose 20). We made this choice mainly due to simplicity of implementation, since our goal in this paper is to study the sequencing of patterns in music rather than to produce complete pieces. In future research, we will revisit the issue of termination of the sequence to study these methods further.

5 Experimental Results

Examples of music composed by our system are presented in Figs. 4 and 5. These results were trained using a piece by Bach, the "Air" from "Orchestral Suite Number 3". Each part of a string quartet was used as a training sequence. One of these was a cello part in the bass clef, which occasionally causes large interval jumps between patterns from different voices. The piece generated does not bear much resemblance to any of the training pieces. Note that the piece mostly obeys basic music theory rules; namely that most of the notes are in the key of the piece, with a few accidental notes, and that the flow between notes is mostly smooth transitions between adjacent notes of the key. One aspect of this example that should also be noted is that the timing of the notes in Fig. 5 is quite complex in a few places. This is largely due to the fact that we do not demand that patterns start on the down-beat. If we did make this constraint, it would simplify the timing by starting patterns on the same part of the beat as in the training sequences. We intend to investigate this issue further in future work.

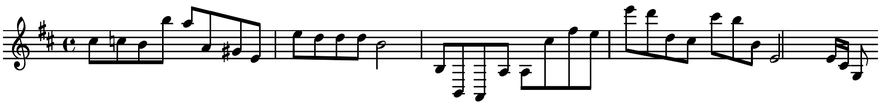


Fig. 4. The piece of this figure was composed using a Bach piece, "Air".

As was discussed previously, our technique expands upon previous work that used Markov chains with each state representing a single note of the scale. An example composition using this model is given in Fig. 6. Note that the progressions from note to note are less smooth, often making large interval jumps. Thus, using longer patterns in the nodes of the Markov chain does provide more continuity in transitions.

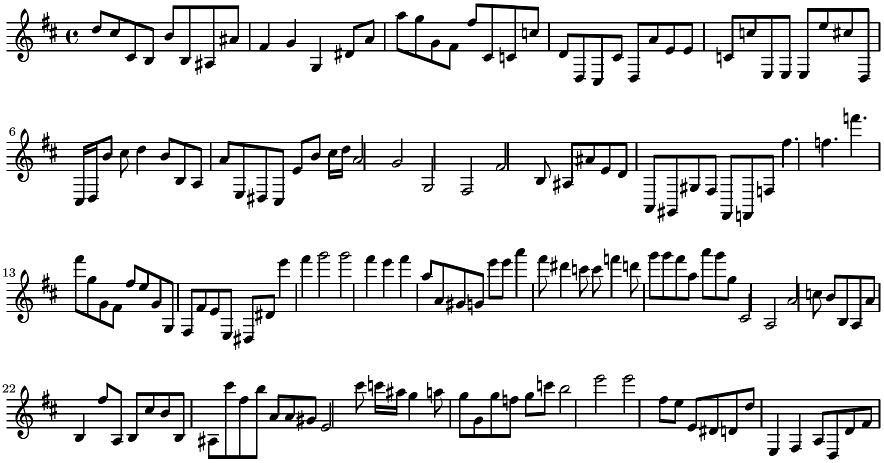


Fig. 5. The piece in this figure is a longer composition based on the same training piece as for Fig. 4 (a Bach "Air").

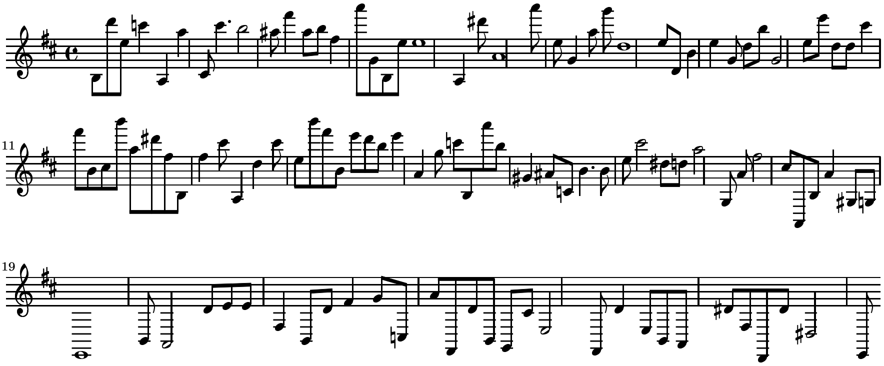


Fig. 6. This piece was composed using the same training data as in Figs. 4 and 5, but using a Markov chain in which each node represents only one note. Note that this composition makes larger interval jumps between consecutive notes, in contrast to the piece of Fig. 5, where the transitions are smoother.

6 Conclusion

In this paper, we have given a new method for algorithmic music composition that takes as input training sequences, such as pieces of classical music, and composes novel music. Our technique first extracts patterns from the training sequences, and then learns from these sequences which patterns can follow others. We construct a

Markov chain in which the nodes represent patterns and the transitions represent sequencing of patterns. We give results of experiments trained on pieces by Bach that demonstrate that our method produces music with smoother transitions than are produced if the nodes of the Markov chain represent only single notes, as has been done in previous work.

References

1. Bainbridge, David, Nevill-Manning, Craig G., Witten, Ian H., Smith, Lloyd A., and McNab, Rodger J., of the Fourth ACM International Conference on Digital Libraries, 1999.
2. Biles, John A., “GenJam: A Genetic Algorithm for Generating Jazz Solos”, In ICMC Proceedings 1994. The Computer Music Association, 1994.
3. Cope, David. “An Expert System for Computer-Assisted Composition”, Computer Music Journal, Vol. 11, No. 4, pp. 30-40.
4. Dannenberg, Roger B., “Music Representation Issues, Techniques, and Systems”, Computer Music Journal, Vol. 17, No. 3, pp. 20-30, 1993.
5. Kim, Youngmoo E., Chai, Wei, Garica, Ricardo, and Vercoe, Barry, “Analysis of a Contour-Based Representation for Melody”, Proc. International Symposium on Music Information Retrieval, Oct. 2000.
6. Lemstrom, Kjell, and Laine, Pauli, “Musical Information Retrieval Using Musical Parameters”, International Computer Music Conference, Ann Arbour, 1998.
7. McCreight, E. M., “A space-economical suffix tree construction algorithm”, Journal of the ACM, Vol. 23, No. 2, pp. 262-272 , 1976.
8. Miranda, Eduardo Reck., *Composing Music with Computers*, Focal Press, Burlington, MA, 2001.
9. Rabiner, L.R., and Juang, B.H., “An Introduction to Hidden Markov Models”, IEEE ASSP Magazine, January 1986.
10. Russel, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2003.
11. Weiner, P., “Linear pattern matching algorithms”, In Proc. 14th Annual Symposium on Switching and Automata Theory, pp. 1-11, 1973.
12. Wiggins, Geraint, Papadopoulos, George, Somnuk, Phon,Amnuaisuk, Tuson, Andrew, “Evolutionary Methods for Musical Composition”, In Proceedings of the CASYS98 Workshop on Anticipation, Music & Cognition, Li`ege, 1998.
13. Witten, Ian H., and McNab, Rodger, “The New Zealand Digital Library: Collections and Experience”, The Electronic Library, Vol. 15, No. 6, 1997.

Analyzing the Performance of Genetically Designed Short-Term Traffic Prediction Models Based on Road Types and Functional Classes

Ming Zhong[1], Satish Sharma[2], and Pawan Lingras[3]

[1] Faculty of Engineering, University of Regina
Regina, SK, Canada, S4S 0A2
zhong1mi@uregina.ca
[2] Faculty of Engineering, University of Regina
Regina, SK, Canada, S4S 0A2
Satish.Sharma@uregina.ca
[3] Dept. of Mathematics and Computing Science
Saint Mary's University
Halifax, NS, Canada, B3H 3C3
Pawan.Lingras@stmarys.ca

Abstract. Previous research indicates that the accuracy of short-term traffic prediction has close relationship with road type, and different models should be used for different types of roads. Road types are characterized by trip purposes and trip lengths. However, research for evaluating short-term model's performance based on functional classes is largely absent. Road's functional class is based on the functionality of individual road in a highway network. In this study, genetically designed locally weighed regression (LWR) and time delay neural network (TDNN) models were used to predict short-term traffic for eight rural roads in Alberta, Canada. These roads belong to different functional classes from four trip pattern groups. The influence of functional classes on the performance of short-term prediction models were studied for each type of roads. The results indicate that not only road trip pattern groups but also functional classes have large influence on the accuracy of short-term traffic prediction.

1 Introduction

Intelligent transportation systems (ITS) are expected to provide cost effective, timely, and flexible solutions for increasingly complicated transportation systems, both in urban and rural areas. Within such systems, advanced traveler information systems (ATIS) are required to monitor real-time traffic situations and be able to predict short-term evolutions. Then the information from an ATIS will be passed to various control devices to maximize the capacity of road networks.

Short-term traffic prediction is one of the most important functionalities of an ATIS. It is evident from the literature that extensive research has been carried out in this area [1-7]. Most of these researches focus on urban roads or rural roads from higher functional classes (e.g., freeways). The reason may be that most congestion

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1133-1145, 2004.

occurs on these roads. For these studies, most of them test the proposed models on one type of road(s). However, ATIS research for all categories of roads is needed as many states plan to implement ITS in both urban and rural areas [8-9]. Due to different traffic characteristics and geographical locations of road sites, the generic applicability of the models developed in the previous research is doubtful. A systematic study is needed to test the general applicability of modeling techniques and their interaction with underlying traffic characteristics.

A few researchers investigated such problem in short-term traffic prediction. Yun *et al.* [10] applied three neural network models, namely the back-propagation model, the finite impulse response (FIR) model, and the time-delayed recurrent model, to three different data sets collected from interstate highways, intercity highways, and urban intersections. It was found that the time-delayed recurrent model outperformed other models in forecasting randomly moving data collected from urban intersections, and the FIR model showed better prediction accuracy for relatively regular periodic data collected from interstate and intercity roads. Their study clearly shows the interactions between underlying traffic characteristics and modeling techniques. However, due to the limited testing sites, it is not possible to conclude if a particular model from their study has a generic applicability to roads from other trip pattern groups and functional classes.

Lingras [11] applied autoregression analysis and time delay neural network (TDNN) to predict daily volumes for nine roads from five trip pattern groups. These trip pattern groups include commuter, regional commuter, rural long-distance, regional recreational, and highly recreational. The study results clearly show that underlying traffic characteristics have large influence on the accuracy of short-term traffic predictions. It was reported that “the prediction errors for predominantly recreational roads are higher than those for predominantly commuter and rural long-distance roads”. He also mentioned that “the errors for different road sites within the same group were similar, and such an observation implies that the conclusions drawn from this study are applicable to the highway categories, instead of specific highway sections”. However, it should be noted that his study predicted daily volumes. Such outputs may not be useful in real world ATIS. Moreover, his conclusion that daily volume prediction models can be applicable to highway categories based on trip pattern groups may not be valid at the hourly level. It is possible that roads from different functional classes have different hourly patterns, and result in different accuracy.

Another drawback of Yun *et al.*’s [10] and Lingras’ study [11] is that they only tested nonlinear models. It is reasonable to use neural networks and autoregressive analysis to model traffic volume time series. However, partitioning traffic volume observations into groups based on their temporal characteristics (e.g., season, day, and hour) shows an increasing linearity, as shown in a previous study [12]. Hence, it is possible that linear models can achieve higher accuracy than nonlinear models based on refined data sets.

This study examines the validity of the conclusion from the previous research that models usually have similar accuracy for the roads from the same trip pattern groups. Sample roads from different functional classes are selected for various trip pattern groups. Models based on different techniques and data structures are developed to investigate the interaction between these factors and resulting accuracy. The influence of both road’s trip pattern group memberships and functional classes on model’s performance is studied.

2 Applied Techniques

Three techniques were used in this study. Locally weighted regression (LWR) and time delay neural networks (TDNN) were utilized as forecasting tools. Genetic algorithms (GAs) were used to choose final inputs from a large number of candidate variables for both LWR and TDNN models. This section gives a brief review to these techniques.

2.1 Locally Weighted Regression Analysis

A variant regression analysis called locally weighted regression was used in this study. Locally weighted regression is a form of instance-based (or memory-based) algorithm for learning continuous mappings from real-valued input vectors to real-valued output vectors. Local methods assign a weight to each training observation that regulates its influence on the training process. The weight depends upon the location of the training point in the input variable space relative to that of the point to be predicted. Training observations closer to the prediction point generally receive higher weights [13]. This is achieved by weighting each data point according to its distance to the query point: a point very close the query gets a weight of one, and a point far away gets a weight of zero.

Model-based methods, such as neural networks and general linear regression, use the data to build a parameterized model. After training, the model is used for predictions and the data are generally discarded. In contrast, “memory-based” methods are non-parametric approaches that explicitly retain the training data, and use it each time a prediction needs to be made. LWR is a memory-based method that performs regression around a point of interest using only training data that are “local” to that point. One benefit of local modeling is that it avoids the difficulty of finding an appropriate structure for a global model. A key idea of local learning is to form a training set for the local model after a query is given. This approach allows to select only relevant samples and to weight them for developing models. After answering the query, the local model is discarded. A new local model is created to answer each query.

Locally weighted regression has been increasingly used in control and prediction. Gorinevsky and Connolly [14] compared several different approximation schemes, such as neural nets, Kohonen maps, radial basis functions, to local polynomical fits on simulated robot inverse kinematics with added noise. They found that local polynomical fits were more accurate than all other methods.

2.2 Time Delay Neural Networks

The variant of neural network used in this study is called time delay neural network (TDNN) [15]. It consists of three layers: input layer, hidden layer, and output layer. Each layer has one or more neurons. There are connections between neurons for processing the information. The input layer receives the information from the outside. The input layer neurons send information to the hidden layer neurons. The neurons in the hidden layer are part of the large internal abstract patterns, which represent the neural network's solution to the problem. The output neurons provide the neural network's response to the input data. The neurons in a given layer of a TDNN can receive input from other neurons on the left delayed by one time interval in the same layer. Such a structure is useful for modeling univariate time series, such as traffic volume, daily temperature, etc.

It is necessary to train a neural network model on a set of examples called the training set so that it adapts to the system it is trying to simulate. Supervised learning is the most common form of adaptation. In supervised learning, the correct output for the output layer is known. Output neurons are told what the ideal response to input signals should be. In the training phase, the network constructs an internal representation that captures the regularities of the data in a distributed and generalized way. The network attempts to adjust the weights of connections between neurons to produce the desired output. The back-propagation method is used to adjust the weights, in which errors from the output are fed back through the network, altering weights as it goes, to prevent the repetition of the error. After the training process is finished, the neural network is tested with an independent data set. If test results are satisfactory, the network is ready for predictions [15].

2.3 Genetic Algorithms for Designing Neural Network and Regression Models

Genetic algorithms (GAs) and simulated annealing (SA) are stochastic search algorithms that attempt to strike a balance between the need to explore the solution space of a problem and the need to focus on the most promising parts of that space. The origin of genetic algorithms (GAs) is attributed to Holland's work [16] on cellular automata. There has been significant interest in GAs over the last two decades [17]. The range of applications of GAs includes such diverse areas as: job shop scheduling, training neural networks, image feature extraction, and image feature identification. Previous research showed that GAs consistently outperformed both classical gradient search techniques and various forms of random search on more difficult problems, such as optimizations involving discontinuous, noisy, high-dimensional, and multimodal objective functions [18].

Hansen, *et al.* [19] used GAs to design time delay neural networks (TDNN), which included the determination of important features such as number of inputs, the number of hidden layers, and the number of hidden neurons in each hidden layer. Hansen, *et al.* [19] applied their networks to model chemical process concentration, chemical process temperatures, and Wolfer sunspot numbers. Their results clearly showed advantages of using TDNN configured using GAs over other techniques including conventional autoregressive integrated moving average (ARIMA) methodology as described by Box and Jenkins [20].

Hansen *et al.*'s approach [19] consisted of building neural networks based on the architectures indicated by the fittest chromosome. The objective of the evolution was to minimize the training error. Such an approach is computationally expensive. Another possibility that is used in this study is to choose the architecture of the input layer using genetic algorithms. Lingras and Mountford [21] proposed the maximization of linear correlation between input variables and the output variable as the objective for selecting the connections between input and hidden layers. Since such an optimization was not computationally feasible for large input layers, GAs were used to search for a near optimal solution. It should be noted here that since the input layer has a section of time series, it is not possible to eliminate intermediate input neurons. They are necessary to preserve their time delay connections. However, it is possible to eliminate their feedforward connections. Lingras and Mountford [21] achieved superior performance using the GAs designed neural networks for the prediction of inter-city traffic.

Lingras *et al.* [22] used GAs to select final independent variables from a large number of candidate input variables for regression models. It was found that GAs designed regression models achieved much higher accuracy than subjectively designed models [22]. The present study uses the same objective function for development of regression and neural network models to predict hourly traffic volumes on various types of roads.

3 Study Data

Currently, Alberta Transportation employs about 350 permanent traffic counters (PTCs) to monitor its highway networks. Hourly traffic volume data collected from these PTCs were used in this study. Hierarchical grouping method proposed by Sharma and Werner [23] was used to classify the roads monitored by these PTCs into groups based on their seasonal traffic distributions. The ratios of monthly average daily traffic (MADT) to annual average daily traffic (AADT) (known as monthly factor $MF = MADT/AADT$) were used to represent the highway sections monitored by these PTCs during the classification. After studying group patterns from 1996 to 2000, five groups were obtained to represent study data. These groups are labeled as the commuter, the regional commuter, the rural long-distance, the summer recreational and the winter recreational groups. Figure 1 shows the grouping results. The commuter group has a flat seasonal pattern due to stable traffic flow across the year. The regional commuter and the rural long-distance group showed higher peaks in the summer and lower troughs in the winter. The summer recreational group has the highest peak in the summer. The largest monthly factor (in August) is about 6 times the smallest monthly factor (in January) for the recreational group. The winter recreational group showed an interesting seasonal pattern – the peak occurred in winter season (from December to March) and the other seasons had low traffic flow.

Eight roads were selected from 4 of these groups: two from the commuter group, two from the regional commuter group, two from the rural long-distance group, and

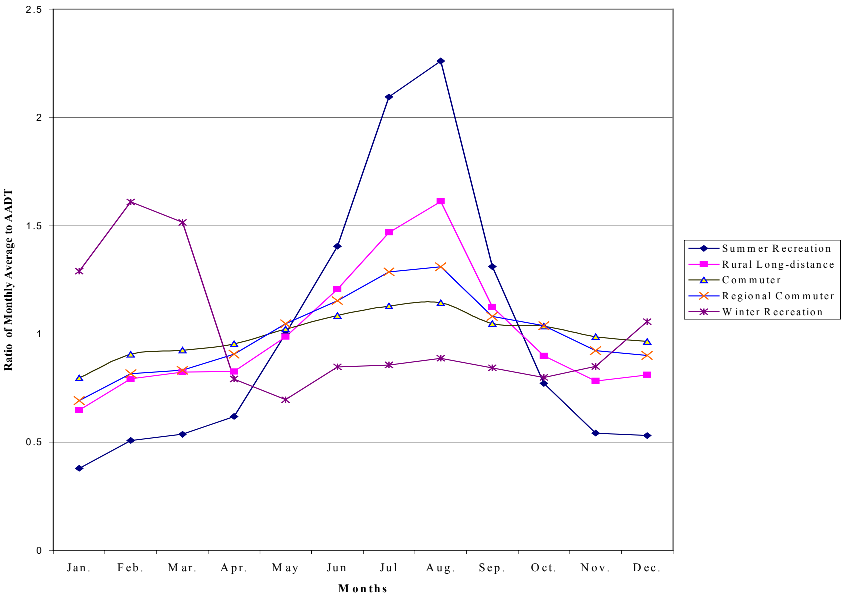


Fig. 1. Hierarchical grouping results of Alberta highway sections

two from the summer recreational group. Because there are not enough data in the winter recreation group, no roads were selected from that group. Table 1 shows the roads selected from different trip-pattern groups, their functional classes, AADT values, location, and training and test data used in this study.

Table 1. Study roads and experimental data from different groups

Trip Pattern Group	Road	Functional Class	AADT	Location	Training Set	Testing Set
Commuter	CM1	Minor Collector	4042	2.5 Km west of 11 and 808 Red Deer	1996 – 1999	2000
	CM2	Principal Arterial	41575	4.6 Km south of 2 and 567 Airdrie	1996 – 1999	2000
Regional Commuter	RC1	Major Collector	3580	2.6 Km west of 3 and 2 Ft Macleod	1996 – 1999	2000
	RC2	Principal Arterial	17889	2 Km north of 2 and 27 Olds	1996 – 1998	2000
Rural Long-distance	RL1	Major Collector	4401	10 Km east of 16 and 40 Hinton EJ	1997 – 1999	2000
	RL2	Minor Arterial	13627	9.62 Km east of 1 and old 1A Canmore	1996 – 1999	2000
Summer Recreation	SR1	Major Collector	2002	6.6 Km south of 1 and 93 Lake Louise	1996 – 1998	2000
	SR2	Minor Arterial	7862	18 Km west of Banff (on TransCanada 1)	1996 – 1998	1999

Figure 2 shows hourly patterns of these roads. For commuter roads (CM1 and CM2), there are two peaks in a day: one is in the morning and the other is in the afternoon. The regional commuter roads RC1 and RC2 also have two peaks in a day, but not as remarkable as those of the commuter roads. The patterns of both RL1 and RL2 have two small peaks. However, the first peak occurred nearly at noon, instead of in the early morning. The summer recreational road SR1 only has one peak occurring nearly at noon. SR2 has two peaks. One is around at noon and the other one occurs in the evening. Most recreational travels on these roads take place in a few hours in the afternoon and early evening.

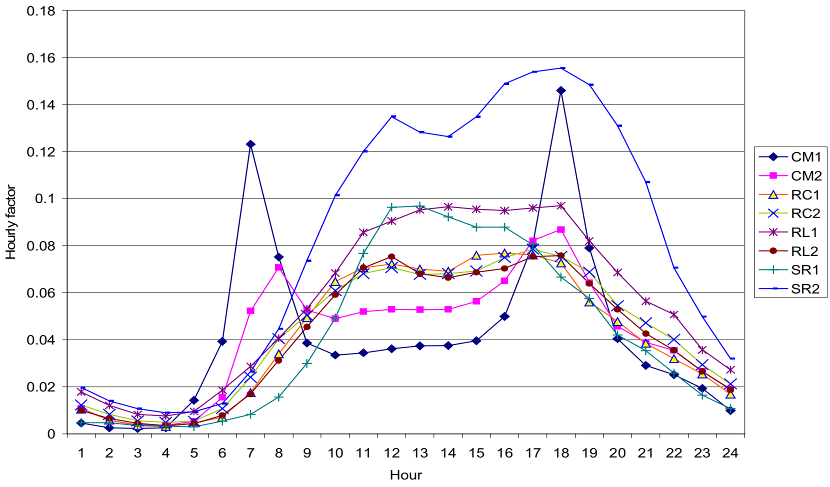


Fig. 2. Hourly pattern for study road sites

Depending on data availability, four or five years data were used in the experiments for each road, as shown in Table 1. The data are in the form of hourly traffic volumes for both travel directions.

4 Study Models

Genetically designed regression and neural network models were designed to forecast the hourly traffic volumes of interest for eight study roads. First one week-long hourly volume time series (7 × 24 = 168 hourly volumes) before the predicted hour were presented to GAs for selecting 24 final inputs. The candidate input set was limited to 168 hourly volumes by assuming that the time series in this period contains all necessary information for short-term prediction of the next hour. The number of final input variables was decided to be 24 because experiments indicated further increasing the number of final inputs led to little or no improvement on model's accuracy. The selection criterion is based on that these 24 input variables have the maximum correlation with the predicted hourly volumes, among all the combinations of selecting 24 from 168 variables (C_{24}^{168}). The connections selected by the genetic algorithm were used to design and implement TDNN and LWR models. The designed models were then used to predict the short-term traffic for all study roads.

In this study, the GA population size was set to 110. GAs were allowed to evolve for 1000 generations. Single point crossover operator was used, and the crossover rate was set at 90%. For mutation, random replacement operator was used and the probability of mutation was set to 1%. These parameters were chosen after experimenting with different values. The best chromosome selected after 1000 generations evolution was used as the final solution of the search.

For TDNN models, there are 168 neurons in the input layer. However, only 24 of them have the connections with the hidden layer neurons. There are 12 neurons in the hidden layer and 1 neuron in the output layer. For regression models, there are 168 independent variables. However, only 24 of them have non-zero coefficients and they are used to predict the dependent variable. Then forecasting models, both regression and neural network models, were applied to predict traffic volumes of interest.

Two types of models were developed in this study. The characteristics of these models are as follows:

- Universal models: This approach involved a single TDNN and a single LWR model for the prediction. The observations for universal models are from 12-hour daytime (8:00 a.m. to 8:00 p.m.) across a year.
- Refined models: This approach involved a TDNN and a LWR model for each hour of each day of the week. The observations for refined models are from the same hours (e.g., 7:00-8:00 a.m.) on the same days of the week (e.g., Wednesdays) in a given season (e.g., July and August).

All the models were trained and then tested. Depending on the model, the number of observations varied. The absolute percentage error (APE) was calculated as:

$$APE = \frac{|actual \quad volume \quad - estimated \quad volume \ |}{actual \quad volume} \times 100 \tag{1}$$

The key evaluation parameters consisted of the average, 85th and 95th percentile errors. These three measures give a reasonable idea of the error distributions by including (e.g., when calculating average errors) or excluding (e.g., when calculating the 85th and the 95th percentile errors) large errors caused by outliers or special events.

5 Results and Discussion

Experiment results are presented in this section. For comparison purpose, the average, 85th and 95th percentile errors for both LWR and TDNN models based on the same data sets are listed in the same table.

Universal models were used to forecast the traffic volume of any hour between 8:00 a.m. and 8:00 p.m. The benefit of universal models is their simplicity of development and implementation. However, the shortcoming was high prediction errors, which was the consequence of using only one model to predict traffic volumes for all hours. Table 2 shows the prediction errors for universal models. TDNN models showed better or comparable performance than LWR models for roads from the commuter, the regional commuter, and the rural long-distance group. For example, for

CM1, the average errors for the TDNN model is 9.68% for the training set, and 12.31% for the test set. Whereas the average errors for the LWR model are all more than 13% for both training and test sets. Locally weighted regression models outperformed neural network models for SR1 and SR2. For SR1, the 95th percentile error for the LWR model is around 46%, whereas the 95th percentile errors are as high as over 70% for both training and test sets. A reason for such difference may be that it is easy for TDNN models to find a similar pattern for roads from the group other than recreational. However, it is difficult to match a similar pattern for recreational roads. Locally weighed regression uses local input variables to predict dependent variables. It is possible that a local model based on the hourly volumes close to the predicted variables could result in more accurate predictions.

Table 2. Errors of traffic prediction using universal Models

Road	Training						Testing					
	Average		85th %		95th %		Average		85th %		95th %	
	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN
CM1	13.40	9.68	21.55	16.55	37.86	25.85	13.15	12.31	21.68	21.96	37.84	34.54
CM2	7.24	4.66	12.10	8.07	22.69	12.20	8.96	6.89	11.69	8.26	20.93	13.07
RC1	9.97	8.85	16.83	15.16	25.17	22.26	10.43	9.26	17.81	16.13	26.10	23.11
RC2	6.72	6.69	11.60	10.53	17.21	16.40	6.46	6.63	11.28	10.80	16.73	16.75
RL1	8.62	8.67	15.32	15.03	22.28	22.75	8.34	8.43	14.66	14.71	21.32	22.30
RL2	8.10	7.30	14.28	12.72	22.76	19.83	7.81	6.99	13.73	11.89	22.08	18.61
SR1	17.18	27.03	28.73	42.09	46.38	79.39	16.15	23.15	27.41	37.79	46.77	73.52
SR2	10.03	10.66	17.41	18.72	28.24	31.01	9.36	9.98	16.53	17.66	25.36	28.74

Refined genetically designed models were applied for short-term traffic prediction for the study roads. The observations for the refined models used here are from same hours of all Wednesdays in the July and August in the study period. One LWR and one TDNN model were developed for each hour between 8:00 a.m. to 8:00 p.m. Table 3 shows the mean average errors, the 85th percentile errors, and the 95th percentile errors for 12 refined hourly models for the eight study roads. It is evident that refined models show higher accuracy. For LWR models, the mean average errors are usually less than 0.5% and the mean 95th percentile errors are below 1.5% for the training sets. Test sets show higher errors. The mean average errors range from 0.58% to 3.17% for different roads. The mean 95th percentile errors range from 1.24% for the commuter road CM2 to 6.88% for the recreational road SR1. TDNN models show higher errors than LWR models. For training sets, mean average errors are usually 4–5% and the mean 95th percentile errors are less than 10%. For test sets, mean average errors are 4-5% for roads with stable patterns (CM2, RC1, RC2, RL2) and about 10% for roads with unstable patterns (CM1, RL1, SR1, SR2). The mean 95th percentile errors for TDNN models are around 10% for roads with stable patterns. For roads with unstable patterns, the mean 95th percentile errors are around 20%.

Table 3. Means errors for refined models

Road	Training						Testing					
	Average		85th %		95th %		Average		85th %		95th %	
	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN	LWR	TDNN
CM1	0.56	4.71	0.97	7.44	1.50	8.40	2.67	9.65	3.95	15.55	5.10	19.59
CM2	0.15	3.04	0.29	5.34	0.38	5.98	0.58	4.49	0.88	6.38	1.24	7.72
RC1	0.46	4.13	0.82	6.36	1.14	6.88	1.74	5.61	2.79	9.03	3.78	12.28
RC2	0.04	3.57	0.07	5.84	0.10	6.54	1.13	5.82	1.60	8.73	2.52	10.35
RL1	0.06	4.34	0.11	6.71	0.15	7.13	2.47	11.58	3.65	17.25	4.61	23.13
RL2	0.28	4.00	0.50	6.66	0.72	7.26	0.87	4.83	1.30	8.25	1.88	9.83
SR1	0.12	5.46	0.21	8.14	0.30	9.58	3.17	9.06	5.02	14.10	6.88	20.12
SR2	0.08	5.05	0.15	8.08	0.19	9.05	1.31	9.37	2.11	13.92	2.55	18.13

It is evident from Table 2 and Table 3 that functional classes have a large influence on accuracy of short-term traffic prediction. For each pattern group, the same models resulted in higher accuracy for the road from the higher functional class. For example, for the recreational road SR2 (minor arterial), the 95th percentile error for the universal LWR model is 28% for the training set and 25% for the test set. Whereas for the recreational road SR1 (major collector), the 95th percentile errors for the LWR model are 46% for both training and test sets. A general conclusion from the previous research [11] is that short-term traffic prediction models usually result in higher accuracy for the roads from more stable traffic pattern groups (e.g., commuter). Based on this conclusion, the short-term traffic prediction errors for commuter roads (e.g., CM1) should be less than those for roads from other groups (e.g., regional commuter, rural long-distance, and recreational). However, the results from Table 2 and Table 3 indicate that this may not be the case. For example, the errors for the commuter road CM1 (minor collector) are larger than those for the regional commuter roads RC1 (major collector) and RC2 (Principal arterial), the rural long-distance roads RL1 (major collector) and RL2 (minor arterial), and the recreational road SR2 (minor arterial), and they are only less than those for SR1 (major collector). It is clear that not only the trip pattern group but also the functional class have large influence on short-term traffic prediction accuracy.

The above results indicate that analyzing short-term traffic patterns is a “key” step in developing models. Different model development strategies should be applied to the roads of different trip-pattern groups, or even different functional classes because they have different short-term traffic patterns. More efforts may be needed to develop better models for roads from lower functional classes than those from higher functional classes to achieve similar accuracy.

6 Conclusion

Literature review indicates that most of the previous ATIS research works tested their models on one type of road(s) [1-7]. The majority of these studies focus on the roads from higher functional classes (e.g., freeways). The generic applicability of the models is not proven. It is necessary to study the interactions between modeling techniques and underlying traffic characteristics of various types of roads to address such an issue. Based on their traffic characteristics and functionalities, roads can be classified into trip pattern groups (e.g., commuter or rural long-distance) and functional classes (e.g., collector or arterial). A few researchers [10-11] tried to study such an issue. However, these studies are limited to a few sample roads or trip pattern group level. No research has been found to study the influence of road's functional class-ship on the model's accuracy.

In this study, roads from different functional classes were selected from four trip pattern groups: the commuter, the regional commuter, the rural long-distance, and the recreational. The influences of both trip pattern groups and functional classes on the accuracy of short-term traffic predictions from LWR and TDNN models are studied. It is evident from the study results that not only road's trip pattern group but also road's functional class have a large impact on the model's performance. For example, the models applied to the roads from stable traffic pattern groups (e.g., commuter) usually result in higher accuracy than those applied to the roads from unstable traffic pattern groups (e.g., recreational). Within each traffic pattern group, the models applied to the road from higher functional class result in the higher accuracy than those applied to the road from lower functional class. Moreover, it is possible that the models applied to a lower functional class road from a stable traffic pattern group (e.g., CM1) result in higher errors than those applied to a higher functional class road from an unstable traffic pattern group (e.g., SR2).

The study results indicate that modeling techniques and underlying data structures have some influences on model's accuracy. When modeling non-linear observation sets of universal models, TDNN models outperform LWR models in most cases. However, for modeling linear observation sets of refined models, LWR models are superior to TDNN models. It is evident from the study results that successful modeling involves knowing the characteristics of modeling techniques and providing proper data structures.

It is advantageous to apply both linear model (e.g., LWR) and non-linear model (e.g., TDNN) to a same data set, as illustrated in this study. Such a mechanism can provide an additional check on the model's accuracy. As shown in this study, LWR models usually result in high accuracy, but they are prone to outliers in data sets. Neural network models are known for their tolerance to interventions in the data. Hence, neural network models based on the same data sets can provide a baseline of accuracy. In this way, both accuracy and stable model performance can be achieved.

Acknowledgements. The authors are grateful towards NSERC, Canada for their financial support and Alberta Transportation for the data used in this study.

References

1. Ahmed, S.A. and Cook, A.R.: Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques. *Transportation Research Record **722***, Transportation Research Board, Washington D.C. (1979) 1-9.
2. Lee, S.S. and Fambro, D.B.: Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. *Transportation Research Record **1678***, Transportation Research Board, Washington D.C. (1999) 179-189.
3. Lu, J.: Prediction of Traffic Flow by an Adaptive Prediction System. *Transportation Research Record **1287***, Transportation Research Board, Washington D.C. (1990) 54-61.
4. Park, B., Messer, C.J., and Urbanik, T.: Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network. *Transportation Research Record **1651**,* Transportation Research Board, Washington D.C. (1998) 39-47.
5. Smith, B.L., and Demetsky, M.J.: Short-Term Traffic Flow Prediction: Neural Network Approach. *Transportation Research Record **1453**,* Transportation Research Board, Washington D.C. (1994) 98-104.
6. Williams, B.M., Durvasula, P.K. and Donald, E.B.: Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average And Exponential Smoothing Models. *Transportation Research Record **1644***, Transportation Research Board, Washington D.C. (1998) 132-141.
7. Zhang, H., Ritchie, S.G., and Lo, Z.P.: Macroscopic Modeling of Freeway Traffic Using an Artificial Neural Network. *Transportation Research Record **1588***, Transportation Research Board, Washington D.C. (1997) 110-119.
8. Minnesota Department of Transportation: Minnesota Guidestar: Shaping Tomorrow's Transportation System Today. <http://www.dot.state.mn.us/guidestar/itsprojects.html> (1999) (June 6, 2002).
9. Washington State Transportation Center: Advanced Traveler Information Systems. <http://www.ota.fhwa.dot.gov/tech/ma/ap71.html> (1999) (Oct. 8, 1999).
10. Yun, S.Y., Namkoong, S., Rho, J.H., Shin, S.W. and Choi, J.U.: A Performance Evaluation of Neural Network Models in Traffic Volume Forecasting. *Mathematical and Computer Modelling*, Vol. 27, (1998) 293-310.
11. Lingras, P.: Traffic Volume Time-Series Analysis According to the Type of Road Use. *Computer-Aided Civil and Infrastructure Engineering* (*15*), (2000) 365-373.
12. Zhong, M., Satish Sharma, and Pawan Lingras: Accuracy of Genetically Designed Models for Predicting Short-term Traffic on Rural Roads. The Proceedings of *the 31st Annual Congress of the Canadian Society of Civil Engineering*, Moncton, New Brunswick (2003).
13. Friedman, J.H.: Intelligent Local Learning for Prediction in High Dimensions. Proceedings of *International Conference on Artificial Neural Networks*, Paris, France (1995).
14. Gorinevsky, D. and Connolly, T.H.: Comparison of Some Neural Network and Scatter Data Approximations: The Inverse Manipulator Kinematics Example. *Neural Computation* 6, pp. (1994) 521-542.
15. Hecht-Nielsen, R.: *Neurocomputing*. Addison-Wesley Pub. Co, Don Mills, Ontario (1990).
16. Holland, J.H.: *Adaptation in Natural and Artificial Systems.* Ann Arbor, University of Michigan Press (1975).
17. Buckles, B.P. and Petry, F.E.: *Genetic Algorithms*. IEEE Computer Press, Los Alamitos, California (1994).
18. Grefenstette, J.: Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, The Institute of Electrical and Electronics Engineers, Inc., 16, No. 1, (1986) 122-128.

19. Hansen, J.V., McDonald, J.B. and Nelson, R.D.: Time Series Prediction with Genetic Algorithm Designed Neural Networks: An Experimental Comparison with Modern Statistical Models. *Computational Intelligence*, 15(3), (1999) 171-184.
20. Box, G. and Jenkins, J.: *Time Series Analysis: Forecasting and Control.* San Francisco, Holden-Day (1970).
21. Lingras, P.J. and Mountford, P.: Time Delay Neural Networks Designed Using Genetic Algorithms for Short Term Inter-City Traffic Forecasting. Proceedings of *the Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Budapest, Hungary (2001) 292-299.
22. Lingras, P.J., Sharma, S.C. and Zhong, M.: Prediction of Recreational Travel using Genetically Designed Regression and Time Delay Neural Network Models. *Transportation Research Record **1805***, Transportation Research Board, Washington, D.C. (2002) 16-24.
23. Sharma, S.C. and Werner, Al.: Improved Method of Grouping Provincewide Permanent Traffic Counters. *Transportation Research Record **815***, Transportation Research Board, Washington D.C. (1981) 12-18.

Nonstationary Time Series Prediction Using Local Models Based on Competitive Neural Networks

Guilherme A. Barreto[1], João C.M. Mota[1], Luis G.M. Souza[2], and Rewbenio A. Frota[2]

[1] Department of Teleinformatics Engineering, Federal University of Ceará
CP 6005, CEP 60455-760, Fortaleza, Ceará, Brazil
{guilherme, mota}@deti.ufc.br
http://www.deti.ufc.br/~guilherme

[2] Instituto Atlântico: Research & Development in Telecom & IT
Rua Chico Lemos, 946, CEP 60822-780, Fortaleza, Ceará, Brazil
{gustavo, rewbenio}@atlantico.com.br

Abstract. In this paper, we propose a general approach for the application of competitive neural networks to nonstationary time series prediction. The underlying idea is to combine the simplicity of the standard least-squares (LS) parameter estimation technique with the information compression power of unsupervised learning methods. The proposed technique builds the regression matrix and the prediction vector required by the LS method through the weight vectors of the *K* first winning neurons (i.e. those most similar to the current input vector). Since only few neurons are used to build the predictor for each input vector, this approach develops local representations of a nonstationary time series suitable for prediction tasks. Three competitive algorithms (WTA, FSCL and SOM) are tested and their performances compared with the conventional approach, confirming the efficacy of the proposed method.

1 Introduction

A scalar time series consists of n observations of a single variable y measured sequentially in the time: $\{y(t), y(t - 1), \dots, y(t - n + 1)\}$. Time series prediction (or forecasting) is the engineering task whose goal is to find mathematical models that supply estimates for the future values of the variable y [2]. This is possible because, in general, successive values of a series are dependent on each other for a period dictated by the underlying process responsible for the generation of the series, which can assume a linear or nonlinear nature. Several approaches for the prediction task have been studied along the years [13], such as the widely used autoregressive (AR) and moving average (MA) models, as well as their combinations in the ARMA and ARIMA models [2], [3]. Among nonlinear models, successful applications using artificial neural networks (ANNs) have been reported elsewhere [4], [5], [10], [13].

In general, existing time series methods can be classified roughly into *global* and *local models* [11]. In global models, a single mathematical model learns the dynamics of the observed series. In local models, the time series is divided into shorter seg-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1146-1155, 2004.

ments, each one characterized by (usually linear) models simpler than the one required by the global approach. The segmentation of the series is usually performed by clustering algorithms, such as the *K-means* [8] or the Self-Organizing Map (SOM) [7], [12]. In this case, a scalar time series is transformed into a set of data vectors by means of a sliding time-window that is formed by a fixed number of consecutive samples of the series. Then, parameters of a given local model are computed using the data associated to the segment it models. The type of model to be used depends on the underlying dynamics of the time series in analysis. Global models are suitable to the prediction of stationary series, while local models are preferred for modeling of nonstationary ones.

This paper introduces a general design technique for building local models for prediction of nonstationary time series that can be used by any type of competitive neural networks. The method is tested with three competitive algorithms, which are evaluated based on their predictive capacity.

The remainder of the paper is organized as follow. Section 2 presents the neural competitive learning algorithms to be used. In Section 3 we discuss the standard linear parameter estimation problem introduces a new technique for local modeling of time series through competitive networks. In Section 4 we report computer simulations involving the methods and networks described in previous sections. The article is concluded in Section 5.

2 Competitive Neural Networks

Competitive learning comprises one of the main classes of unsupervised ANNs, where only a neuron or a small group of neurons, called *winning neurons*, are activated according to the degree of proximity of their weight vectors to the current input vector [4], [6], [10]. This type of algorithm is used in tasks of pattern recognition and classification, such as clustering and vector quantization. In these applications, the weight vectors are called the *prototypes* of the set input patterns. In the simplest competitive algorithm, known as WTA (*Winner-Take-All*), only one neuron has the weights updated. The training can be described in two basic steps:

1. Search for the winning vector, $i^*(t)$, associated with the input vector $\mathbf{x}(t)$:

$$i^*(t) = \arg\min_{\forall i} \left\| \mathbf{x}(t) - \mathbf{w}_i(t) \right\| \ . \tag{1}$$

2. Updating the weight vectors, $\mathbf{w}_i(t)$, of the winning vector,

$$\Delta \mathbf{w}_{i^*}(t) = \alpha(t)\left[\mathbf{x}(t) - \mathbf{w}_{i^*}(t)\right], \tag{2}$$

where $0 < \alpha(t) < 1$ it is the learning rate that should decrease with the time for convergence purposes. In this paper, we adopt $\alpha(t) = \alpha_0(1- t/T)$, where α_0 is the initial value of $\alpha(t)$ and T is the maximum number of training iterations. A limitation of the WTA is its high sensitivity to weight initialization, a problem that leads to the occurrence of *dead units*, i.e., neurons never selected as winners. To avoid this, simple modifications to the original WTA algorithm have been proposed to give chance to all neurons to become a winner at some stage of the training. The first algorithm of interest, called *Frequency-Sensitive Competitive Learning* (FSCL) [1], modifies Equation (1) by introducing a weighting factor that penalizes strongly those neurons that have been selected too often:

$$i^*(t) = \arg\min_{\forall i}\{f_i(t)\cdot\|\mathbf{x}(t)-\mathbf{w}_i(t)\|\} \;. \tag{3}$$

where $f_i(t) = [c_i/t]^z$, so that c_i is the number of times a neuron i is selected as winner until iteration t, and $z \geq 1$ is a constant exponent. The adjustment of the weights follows in accordance with Equation (2).

The second algorithm is the well-known *Self-Organizing Map* (SOM) [6], which modifies Equation (2) so that the weights of the neurons in the neighborhood of the winning neuron are also adjusted:

$$\Delta\mathbf{w}_{i*}(t) = \alpha(t)h(i^*,i;t)\left[\mathbf{x}(t)-\mathbf{w}_{i*}(t)\right] \;. \tag{4}$$

where $h(i^*,i;t)$ is a Gaussian weighting function defined by:

$$h(i^*,i;t) = \exp\left(-\frac{\|\mathbf{r}_i(t)-\mathbf{r}_{i*}(t)\|^2}{\sigma^2(t)}\right) \;. \tag{5}$$

in which σ defines the width of the neighborhood of neuron i, while $\mathbf{r}_i(t)$ e $\mathbf{r}_{i*}(t)$ are, respectively, the positions of the neurons i and i^* in the SOM. The variable $\sigma(t)$ should also decreases with time as $\sigma(t) = \sigma_0(1- t/T)$, where σ_0 is its initial value.

3 Building Local Models via Competitive Neural Networks

Currently, build local models for time series prediction via competitive ANNs is based on the training data set [7], [8], [12]. In this formulation, the network is used only to separate the input vectors per neuron. It is assumed that the input vector in the instant t is given by:

$$\mathbf{x}(t) = [y(t+1)\,|\,y(t) \quad \cdots \quad y(t-n_y+1)]^T \tag{6}$$

$$= [x_1(t) \quad x_2(t) \quad \cdots \quad x_{n_y+1}(t)]^T \tag{7}$$

where $n_y > 1$ is the length of the window used to built the input vectors from consecutive samples of the time series. For a series with N samples, it is possible to get N - n_y input vectors. After training is completed, the same set of input vectors used for training is presented once again in order to separate them per neuron. No weight adjustment is performed at this stage. We denote by $\mathbf{x}^i(t)$ an input vector $\mathbf{x}(t)$ for which the neuron i is the winner. Then, to each neuron i we associate a *linear* AR models whose parameters are computed using only the corresponding vectors $\mathbf{x}^i(t)$.
An autoregressive linear model (AR) of order n_y is represented by:

$$\hat{y}(t+1) = a_0 + \sum_{j=1}^{n_y} a_j y(t-j+1) \tag{8}$$

where a_j are the coefficients of the model. The prediction errors (or residuals) $e(t) = y(t) - \hat{y}(t)$ are used to evaluate the accuracy of the model by means of the *Normalized Root Mean Square Error* (NRMSE):

$$NRMSE = \sqrt{\frac{\sum_{k=1}^{N} e^2(k)}{N \cdot \sigma_y^2}} = \sqrt{\frac{\hat{\sigma}_e^2}{\sigma_y^2}} \tag{9}$$

where $\sigma_y{}^2$ is the variance of the original series, $\hat{\sigma}_e^2$ is the variance of the residuals and N is the length of the sequence of residuals. To calculate the coefficients a_j we use the well-known *Least-Squares* (LS) method [4], [5], [10]. Thus, the coefficients of the AR model associated with neuron i are computed as follows:

$$\mathbf{a}^i = \left(\mathbf{R}_i^T \mathbf{R}^T\right)^{-1} \mathbf{R}_i^T \mathbf{p}_i \tag{10}$$

where the vector of predictions p_i and the regression matrix R_i are built from the vectors $\{\mathrm{x}^i(t_1), \mathrm{x}^i(t_2), \ldots, \mathrm{x}^i(t_{\mathrm{N}i})\}$ for which the neuron i was the winner at instants $\{t_1, t_2, \ldots, t_{\mathrm{N}i}\}$, respectively. By means of Equations (7) and (8), we have that:

$$\mathbf{p}_i = [x_1^i(t_1) \quad x_1^i(t_2) \quad \cdots \quad x_1^i(t_{N_i})]^T \,. \tag{11}$$

$$\mathbf{R}_i = \begin{pmatrix} 1 & x_2^i(t_1) & \cdots & x_{n_y+1}^i(t_1) \\ 1 & x_2^i(t_2) & \cdots & x_{n_y+1}^i(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_2^i(t_{N_i}) & \cdots & x_{n_y+1}^i(t_{N_i}) \end{pmatrix} . \tag{12}$$

Once the coefficients a^i are computed by Equation (10), we use Equation (8) to predict new values for the time series.

In this paper, we propose the use of the weight vectors of a competitive ANN to build local predictors, since these weights constitute the *prototypes* of the training data vectors. For this purpose, at each time t, we need to find K neurons, $\{ i_1^*(t), i_2^*(t), \ldots, i_K^*(t) \}$, whose weight vectors are the closest to the input vector:

$$i_1^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \,. \tag{13}$$

$$i_2^*(t) = \arg\min_{\forall i \neq i_1^*} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \,. \tag{14}$$

$$\vdots \qquad \qquad \vdots$$

$$i_K^*(t) = \arg\min_{\forall i \neq \{i_1^*, \cdots, i_{K-1}^*\}} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \,. \tag{15}$$

where $i_1^*(t)$ is the first winning neuron, $i_2^*(t)$ is the second winning neuron, and so on, until the K-th winning neuron, $i_K^*(t)$. Then, we apply the LS method to the weight vectors of these K neurons to compute the coefficients of Equation (8). Thus, the prediction vector p and the regression matrix R, are now given by:

$$\mathbf{p} = [w_{i_1^*,1}(t) \quad w_{i_2^*,1}(t) \quad \cdots \quad w_{i_K^*,1}(t)]^T \tag{16}$$

$$\mathbf{R} = \begin{pmatrix} 1 & w_{i_1^*,2}(t) & w_{i_1^*,3}(t) & \cdots & w_{i_1^*,n_y+1}(t) \\ 1 & w_{i_2^*,2}(t) & w_{i_2^*,3}(t) & \cdots & w_{i_2^*,n_y+1}(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & w_{i_K^*,2}(t) & w_{i_K^*,3}(t) & \cdots & w_{i_K^*,n_y+1}(t) \end{pmatrix}. \tag{17}$$

where w_{ij} corresponds to the j-th element of the weight vector of neuron i. For the present case, $i \in \{ i_1^*(t), i_2^*(t), \ldots, i_K^*(t) \}$ and $j = 1, \ldots, n_y + 1$. The index i in **p** and **R** is eliminated because now we do not refer anymore to a local model associated to neuron i, but rather to a local model associated to the K winning neurons, for each vector $\mathbf{x}(t)$. The main advantages of the proposed method are listed below:

- **Lower computational cost**: The conventional method trains the ANN, then it separates the data per neuron i, and finally calculate the coefficients of the various local AR models. In the proposed method, only a single local model is built using the weight vectors of K winning neurons. A data separation stage is not needed anymore.
- **Greater numerical stability**: When we use only K neurons out of N ($K >> N$) available, the require matrix inversion required by Equation is performed on a regression matrix of lower dimension than that of the traditional method.
- **Greater robustness**: In competitive ANNs, the weight vector of neuron i converges to the centroid of the cluster of input vectors for which this neuron was the winner [4], [6], [10]. It is well known that "averaging" serves to alleviate the effects of noise, so that local models built from the weight vectors will turn to be more robust to distortions in the input vectors.
- **Greater generality**: Previous attempts to building local models are based on properties that are inherent only to the SOM algorithm [7], [12]. Any competitive ANN, not only for the SOM, can use the methodology proposed in this paper.

3.1 Nonstationary Time Series

The time series used in the simulations is shown in Fig. 1. This series was generated from the Mackey-Glass differential equation with time delay, t_d [9]:

$$\frac{dy(t)}{dt} = \gamma_{t_d} = -0.1y(t) + \frac{0.2y(t-t_d)}{1+[y(t-t_d)]^{10}} \tag{18}$$

where $y(t)$ is the value of the time series at time t. Equation (18) models the dynamics of the production of white blood cells in patients with leukemia. A nonstationary time series is obtained from the composition of three stationary modes of operation of Equation (18), termed A, B and C, corresponding to different delays t_d = 17, 23 and 30. After iterating for 400 instants of time in mode A, the dynamics from the series is switched to a mixture of modes A, B and C, given by:

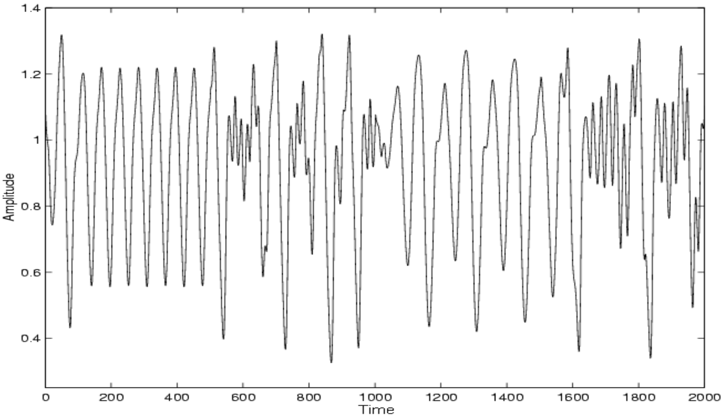


Fig. 1. The simulated nonstationary time series used in this paper.

$$\frac{dy(t)}{dt} = a\gamma_{17} + b\gamma_{23} + c\gamma_{30} \tag{19}$$

so that a = 0.6, b = 0.3 and c = 0.1. The system runs in this combined mode for the following 400 instants of time, when then the dynamics is switched to mode B (t = 801, ..., 1200). After that, the system is changed to a new mixture of modes, for which a = 0.2, b = 0.3 and c = 0.5 until it reaches t = 1600. Finally, from t = 1601, ..., 2000, the system runs in mode C. The first 1500 samples of the generated series are for training the networks and the remaining samples are used for testing[1].

4 Simulations

The simulations aim to evaluate the performance of the proposed local prediction method, comparing the results obtained by the three competitive ANNs presented in Section 2 with the conventional approach by [7] and [12]. For this purpose, we adopted the one-step-ahead prediction task, in which we estimate only the next value $y(t+1)$ of the time series. In this kind of prediction, the input of the network consists of actual values of the series, not the estimated ones. In other words, the predictions are not fed back to the input of the network. For all simulations, we used 50 neurons, α_0 = 0.9, σ_0 = 25 and T=10^4.

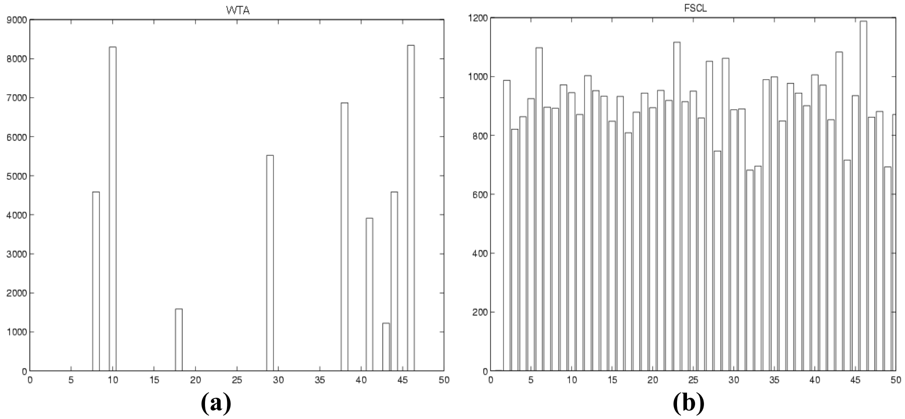
The first set of simulations assess the quality of learning of the WTA, FSCL and SOM algorithms in terms of the number of dead units they generate. This issue is very important, since the proposed method depends on the distribution of the weight vectors in the data space: the better the clustering of the input vectors, the better will be

[1] To discretize Equation (19) we used the simple Euler method:

$$\frac{dy(t)}{dt} \approx \frac{y(t-1) - y(t)}{\Delta} \tag{20}$$

where Δ is the discretization step. In this work, it was adopted $\Delta = 1$.

the predictive capacity of the local prediction model. This quality can be roughly evaluated by counting of the number of times each neuron is selected the winning neuron during the training of a certain ANN. An approximately uniform distribution of the number of victories among neurons is indicative of good learning. As expected, the worst performance was presented by the WTA (Fig. 2a), while the FSCL had the best performance (Fig. 2b). For this simulation, we set $n_y = 5$.



(a) **(b)**

Fig. 2. Distribution of victories per neuron for (a) the WTA and (b) the FSCL.

The second test assesses the "speed" of knowledge acquisition by the networks. This issue can be evaluated by measuring the influence of the length of the training set on the prediction error. Each ANN is trained using only the first 100 samples of the series. This value is increased by 100 samples, until reaching the maximum length of 1500 samples. The time series for testing remains the same for all cases. For this simulation, we set $K = 20$ and $n_y = 5$. For each length of the training set, each ANN is trained 3 times and the average value of the prediction error for the test set is calculated. The results are shown in Fig. 3a in which we also verify that the WTA performed worse than the FSCL and the SOM algorithms.

The third test evaluates the influence of the memory parameter (n_y) on the final prediction error, for a fixed value of $K = 20$. Fig. 3b shows a typical result for the proposed method using the SOM. In this figure, the value of the error decays until reaching its minimum at $n_y = 5$, starting to grow for $n_y \geq 6$. The same behavior of the error is observed for the others two ANNs, varying only the point where the error reaches its minimum: WTA ($n_y = 11$) and FSCL ($n_y = 7$).

The fourth test evaluates the influence of the number of winning neurons (K) on the final value of the prediction error during the testing stage. We fixed $n_y = 5$ for all neural networks. Fig. 4a shows a typical result for the FSCL algorithm. In this figure, the value of the error presents an initial irregular pattern, until stabilizing by $K = 20$. From this value on, it is not worth increasing K because no substantial reduction of the error is observed. The same general behavior is observed for the FSCL and the SOM, varying only the point where the error seems to stabilize: WTA ($K = 25$) and SOM (K $=17$). The results in Figs. 3 and 4 suggest strongly that good training imply lower values for the parameters n_y and K. Thus, among the three competitive ANNs we tested the proposed local method, the SOM and FSCL algorithms performed much better than the WTA algorithm.

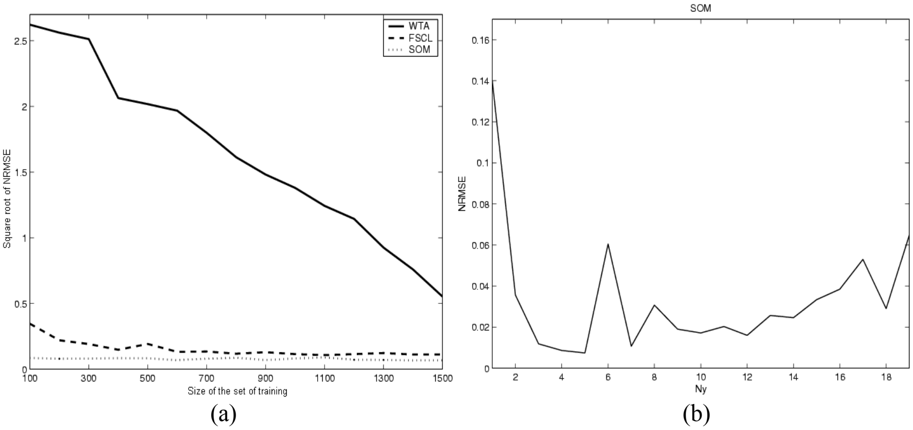


Fig. 3. Influence of (a) the size of the training set and (b) the order n_y on the prediction error.

A typical sequence of predicted values is shown in Fig 4b, in which the solid line represents the actual time series, while small open circles represent the estimated values. For this simulation, we set K = 17 and n_y = 5. As can be seen, the obtained estimates are very similar to the actual values.

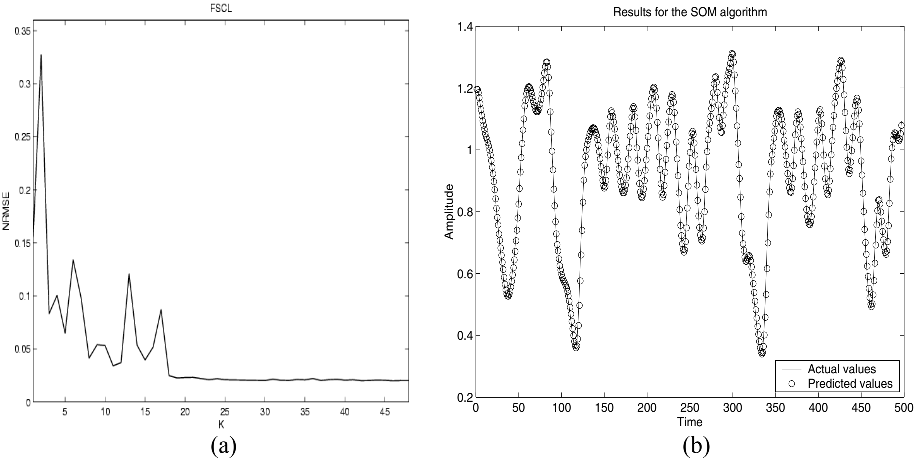


Fig. 4. (a) Influence of the number of winning neurons K on the prediction error for the FSCL. (b) A time series predicted by the SOM algorithm using the proposed approach.

Numerical values of the prediction error comparing the method proposed in this paper with the one in [12] are shown in Table 1. In this table, SOM-K denotes the proposed local method based on K winning weight vectors of the SOM, while SOM-D refers to the local model based the original data vectors. Ten runs of training and testing were carried out, for which the maximum, minimum and average values, as well as the variance of the prediction error were computed. The results show that the SOM-K approach performed better than the SOM-D.

Table 1. NRMSE values for the proposed approach (WTA, FSCL and SOM-K) and the conventional method (SOM-D) described in [12].

Neural Network	**NRMSE**			
	Minimum Value	***Maximum Value***	***Mean Value***	***Standard deviation***
WTA	0.1453	0.4358	0.2174	0.0872
FSCL	0.0091	0.0177	0.0114	0.0032
SOM-K	**0.0055**	**0.0058**	**0.0057**	**0.0002**
SOM-D	0.0076	0.0132	0.0093	0.0027

5 Conclusion

In this paper it is proposed a new methodology for applying competitive neural networks to nonstationary time series prediction tasks. The traditional approach uses competitive ANNs only to separate the input data vectors per neuron of the network. In this approach local autoregressive models are built using only the subset of data vectors associated with a given neuron. The method proposed in this paper suggests the use of the weight vectors of the *K* winning neurons found for the current input vector. Since only few neurons are used to build the predictor for each input vector, this approach develops local representations of a nonstationary time series suitable for prediction tasks.

The advantages of the proposed technique are its greater generality, lower computational cost, greater robustness to noise and greater numerical stability. Three competitive algorithms (WTA, FSCL and SOM) were tested and their performances compared with the conventional approach, confirming the efficacy of the proposed method. For future work, the authors aim to compare the performance of the methods presented in this paper with supervised neural networks, such as the MLP and the RBF, in time series prediction tasks.

Acknowledgments. The authors thank the financial support of CNPq (DCR grant 305275/02-0) and Instituto Atlântico Research and Development Center in Telecom & IT.

References

1. Ahalt, S., Krishnamurthy, A., Cheen, P. and Melton, D.: Competitive learning algorithms for vector quantization, *Neural Networks* 3 (1990) 277–290

2. Box, G. and Jenkins, G.: *Time series analysis, forecasting and control*, Holden-Day, San Francisco (1970)
3. Barreto, G. A. and Andrade, M. G.: Robust bayesian approach for AR(p) models applied to streamflow forecasting, *Journal of Applied Statistical Science* 13 (2004)
4. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, Macmillan/IEEE Press (1994)
5. Haykin, S. and Principe, J.: Making sense of a complex world: Using neural networks to dynamically model chaotic events such as sea clutter, *IEEE Signal Processing Magazine* **15** (1998) 66–81
6. Kohonen, T.: *Self-Organizing Maps*, 2nd extended edn, Springer-Verlag, Berlin, Heidelberg (1997)
7. Koskela, T., Varsta, M., Heikkonen, J. and Kaski, S.: Time series prediction using recurrent SOM with local linear models, *International Journal of Knowledge-based Intelligent Engineering Systems* **2** (1998) 60–68
8. Lehtokangas, M., Saarinen, J., Kaski, K. and Huuhtanen, P.: A network of autoregressive processing units for time series modeling, *Applied Mathematics and Computation* **75** (1996) 151–165
9. Mackey, M. C. and Glass, L.: Oscillations and chaos in physiological control systems, *Science* **197 (1977)** 287–289
10. Principe, J. C., Euliano, N. R. and Lefebvre, W. C.: *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley & Sons (2000)
11. Principe, J., C., Wang, L. and Motter, M. A.: Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control, *Proceedings of the IEEE*, 86 (1998) 2240-2258
12. Vesanto, J.: Using the SOM and local models in time series prediction, *Proceedings of the Workshop on Self-Organizing Maps*, (WSOM'97), Espoo, Finland (1997) 209–214
13. Weigend, A. and Gershefeld, N.: *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading (1993)

Predictive Modeling and Planning of Robot Trajectories Using the Self-Organizing Map

Guilherme A. Barreto[1] and Alu’zio F.R. Araj□o [2]

[1] Universidade Federal do Cear□(UFC)
Departamento de Engenharia de Teleinform□tica
Campus do Pici, Centro de Tecnologia, Fortaleza, CE, Brazil
guilherme@deti.ufc.br
http://www.deti.ufc.br/~guilherme

[2] Universidade Federal de Pernambuco (UFPE)
Centro de Inform□tica - CIn
Departamento de Sistemas da Computa□□o
Av. Professor Lu’s Freire, s/n, Cidade Universit□ria
50740-540, Recife, PE, Brazil
Recife/PE, Brazil
aluizioa@cin.ufpe.br

Abstract. In this paper, we propose an unsupervised neural network for prediction and planning of complex robot trajectories. A general approach is developed which allows Kohonen's Self-Organizing Map (SOM) to approximate nonlinear input-output dynamical mappings for trajectory reproduction purposes. Tests are performed on a real PUMA 560 robot aiming to assess the computational characteristics of the method as well as its robustness to noise and parametric changes. The results show that the current approach outperforms previous attempts to predictive modeling of robot trajectories through unsupervised neural networks.

1 Introduction

Prediction of robot trajectories is a recent research topic in the Neural Network literature [1], [2], [5], [6], [7]. In this context, prediction is understood as the capacity of an artificial neural network (ANN) to determine the next state (configuration) of robot with respect to a particular trajectory. The training process of this ANN makes it able to execute complex robotic control tasks, such as autonomous trajectory planning, avoidance of obstacles in an environment, and nonlinear predictive control.

The early unsupervised neural approaches to learning robotic trajectories employed *static* ANNs [12], [13], i.e., models that are understood as systems without memory. In this situation, the neural algorithm has to learn a mapping considering input-output patterns that occur at the same instant of time, such as forward and inverse kinematics. However, this approach has limited applicability to robotic tasks which require the learning of both spatial and temporal relationships, such as trajectory planning. Static

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1156-1165, 2004.

ANNs can also be used to trajectory planning, but in this case, the temporal order of the trajectory states is not learned by the network itself, but rather predefined by the robot operator.

This paper presents a neural network model based on the Self-Organizing Map (SOM) algorithm [10] to predictive modeling of complex robot trajectories. In this approach, the neural network automatically learns the temporal order of the trajectory states through associative memory mechanisms. The proposed model is then used to plan and control trajectories of a real 6-DOF PUMA 560. Moreover, the tests were also designed to estimate appropriate parameters and to evaluate the model□s robustness to input noise and parametric changes.

The remainder of this paper is organized as follows. A brief summary on the SOM is presented in Section 2, in which we show how it is used to build static input-output mappings. In Section 3, a temporal associative schema, which extends the SOM algorithm to learn dynamical mappings, is described. Such a strategy is employed to plan and control trajectories of the PUMA 560 and the implementation of this method is discussed in Section 4. Section 5 concludes the paper.

2 Self-Organizing Maps and Dynamic Mappings

The SOM is an unsupervised neural network algorithm developed to learn neighborhood (spatial) relationships of a set of input data vectors. Each neuron i has a weight vector $\mathbf{w}_i \in \Re^n$ with the same dimension of the input vector $\mathbf{x} \in \Re^n$, and all neurons are usually arranged in a two-dimensional array, called output layer. The SOM learning algorithm can be summarized in two basic steps:

1. Find the winning neuron at time t: $i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$ (1)

2. Then, update the weight vectors: $\Delta \mathrm{w}_i(t) = \eta(t)h(i^*, i; t)[\mathrm{x}(\mathrm{t}) - \mathrm{w}_i(t)]$ (2)

where $0 < \eta(t) < 1$ is the learning rate and $h(i^*, i; t) = \exp(-\| \mathbf{r}_{i^*}(t) - \mathbf{r}_i \|^2 / 2\sigma^2(t))$ is a Gaussian neighborhood function in which $\mathbf{r}_i(t)$ and $\mathbf{r}_{i^*}(t)$ denote the positions of the neurons i and i^* in the output array. For the sake of convergence, the parameters $\eta(t)$ and $\sigma(t)$ should decrease in time, for example, in a linear basis: $\eta(t) = \eta_0(1-t/T)$ and $\sigma(t) = \sigma_0(1-t/T)$, where η_0 and σ_0 are the initial values of $\eta(t)$ and $\sigma(t)$, respectively. T denotes the maximum number of training iterations. Another common choice for the neighborhood function is the rectangular one (also called bubble): $h(i^*, i; t) = 1$, if $i \in V_{i^*}(t)$, and $h(i^*, i; t) = 0$, otherwise. The set $V_{i^*}(t)$ contains all neurons in the neighborhood of the winning neuron i^* at time t. As in the Gaussian case, the size of $V_{i^*}(t)$ should also decay in time for convergence purposes.

The work by [13] extended the SOM network to learn mappings from input-output pairs of static patterns. Such static mappings are often described by:

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{u}(t)) \tag{3}$$

where $\mathbf{u}(t) \in \Re^n$ denotes he input vector and $\mathbf{y}(t) \in \Re^m$ is the output vector. More recently, the work by [4] generalized the model of Walter and Ritter in order to encode dynamic mappings, such as those described in [9]:

$$\mathbf{y}(t+1) = \mathbf{f}[(\mathbf{y}(t), \ldots, \mathbf{y}(t-n_y+1); \mathbf{u}(t), \ldots, \mathbf{u}(t-n_u+1)] \tag{4}$$

where n_u and n_y are the orders of the input and output memories. Equation (4) indicates that the output at time instant t+1 depends on the n_y past outputs and n_u past inputs. Usually, the mapping **f**(.) is nonlinear and unknown.

In order to establish temporal associations between consecutive patterns in a temporal sequence, neural networks need to retain information about past sequence items [3]. Such a retention mechanism, called *short-term memory* (STM), encodes temporal order and/or temporal dependencies between successive sequence patterns. STM can be implemented by different strategies, the simplest being the so-called *tapped delay line*, understood as a sliding □time window□ over the input sequence within which a number of successive samples are concatenated into a single pattern vector of higher dimensionality. In the following, we use delay lines as the STM mechanism to allow the SOM to learn input-output dynamical mappings.

3 Temporal Associative Memory

In order to approximate **f**(.), the input and output vectors of a time series {**u**(t), **y**(t)}, t = 1, □ , N are organized into a single input vector to presented to the SOM. The first component of $\mathbf{x}^{in}(t)$ represents the actual input information of the mapping whereas the second piece, $\mathbf{x}^{out}(t)$, corresponds to the desired output information of the same mapping. The input and weights vectors are then redefined as

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^{in}(t) \\ \mathbf{x}^{out}(t) \end{pmatrix} \quad \text{and} \quad \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^{in}(t) \\ \mathbf{w}_i^{out}(t) \end{pmatrix} \tag{5}$$

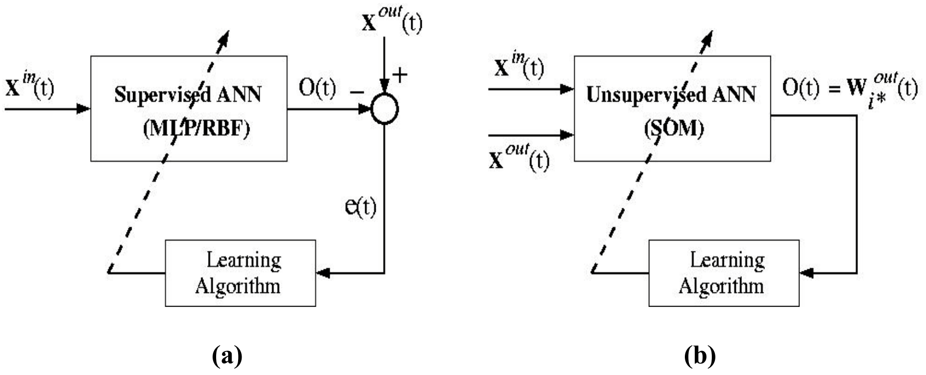
During the training phase, the winning neurons are found based solely on the input component of the extended input vector **x**(t):

$$i^*(t) = \arg\min_{\forall i} \left\| \mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t) \right\| \tag{6}$$

However, the weight updates consider both parts of the input vector:

$$\Delta \mathbf{w}_i^{in}(t) = \eta(t) h(i^*, i; t)[\mathbf{x}_i^{in}(t) - \mathbf{w}_i^{in}(t)] \tag{7}$$

$$\Delta \mathbf{w}_i^{out}(t) = \eta(t) h(i^*, i; t)[\mathbf{x}_i^{out}(t) - \mathbf{w}_i^{out}(t)] \tag{8}$$



(a) (b)

Fig. 1. Difference between (a) supervised and (b) unsupervised (MATQV) learning of input-output mappings.

Thus, by means of Equations (5)-(8) the SOM model learns to associate the input signals with the output ones, while simultaneously performing vector quantization of the input and output spaces. Bearing that in mind, this technique is called *Vector Quantized Temporal Associative Memory* (VQTAM). The VQTAM may be used for approximating different types of mapping depending on the nature of the vectors $\mathbf{x}^{in}(t)$ and $\mathbf{x}^{out}(t)$. For example, the VQTAM formulation of the system defined in Equation (4) is given by

$$\mathbf{x}^{in}(t) = [\mathbf{y}(t), \ldots, \mathbf{y}(t-n_y+1); \mathbf{u}(t), \ldots, \mathbf{u}(t-n_u+1)]^T \quad \text{and} \quad \mathbf{x}^{out}(t) = \mathbf{y}(t+1) \tag{9}$$

Once the mapping is learned, the VQTAM may be used to estimate the output values of this mapping by the equation stated below

$$\ddot{\mathbf{y}}(t+1) = \mathbf{w}_{i*}^{out}(t) \tag{10}$$

where the winning neuron $i^*(t)$ is determined as in Equation (6). The prediction process is repeated M times until a time series of estimated values is available.

Training the VQTAM is characterized by simultaneous presentation of the vectors $\mathbf{x}^{in}(t)$ and $\mathbf{x}^{out}(t)$ and the absence of an explicit computation of an error signal (Fig. 1b). On the other hand, a supervised learning algorithm, such as those used in RBF and MLP networks, employ an error to guide the training, so that only the vector $\mathbf{x}^{in}(t)$ is used as input of the network and the vector $\mathbf{x}^{out}(t)$ is the desired output (Fig. 1a) needed to compute the error.

3.1 Predictive Modeling of Robot Trajectories

The VQTAM makes it possible to learn and to reproduce complex mappings, such as those responsible for the generation of robot trajectories. In particular, the focus of this study is the prediction of joint angles associated with a given robot trajectory of the 6-degree-of-freedom PUMA 560 robot, and their posterior reproduction for trajectory planning purposes. Let the vector $\boldsymbol{\theta}(t) = [\theta_1(t) \ldots \theta_6(t)]^T$ represent the vector of joint angles at time *t*. In this context, Equation (4) reduces to

$$\boldsymbol{\theta}(t+1) = \mathbf{f}[(\boldsymbol{\theta}(t),\ldots,\boldsymbol{\theta}(t-n_\theta+1)] \tag{11}$$

where n_θ is the model order, also called memory parameter. This formulation allows the SOM to learn the kinematics of the manipulator while simultaneously encoding the temporal order of the trajectory states. In this case, the vectors $\mathbf{x}^{in}(t)$ and $\mathbf{x}^{out}(t)$ are the following:

$$\mathbf{x}^{in}(t) = [\boldsymbol{\theta}(t),\ldots,\boldsymbol{\theta}(t-n_\theta+1)]^T \quad \text{and} \quad \mathbf{x}^{out}(t) = \boldsymbol{\theta}(t+1) \tag{12}$$

where the determination of the winning neuron follows Equation (9) and the weight adjustments are determined by Equations (10) and (11). The estimate for the next vector of joint angles is then given by

$$\ddot{\boldsymbol{\theta}}(t+1) = \mathbf{w}_{i*}^{out}(t) \tag{13}$$

Such an estimate can then used as setpoint for autonomous planning and control of the manipulator (Fig. 2). In the experiment to be described in the next section, the robot operator defines the initial state of the trajectory for $t = 0$, and the neural network generates the next state. Such a state is sent to the robot that moves itself to the desired position. The new configuration of the joints of the PUMA is then measured and fed back to the network that produces the next state. This procedure is repeated until the end of the trajectory is reached. Due to the inherent perturbations, very often the robot moves to a neighborhood of the target position, thus the neural networks has to be robust to these perturbations. That is, the generated responses should be stable, i.e., close enough to the states of the learned trajectory.

It is also worth emphasizing the differences between the predictive approach to autonomous trajectory planning and the conventional *look-up table* method [11]. Usually, a trajectory is taught to the robot by the well-known *walk-through* approach: the operator guides the robot by means of a teach-pendant through the sequence of desired arm positions [8]. These positions are then stored for posterior reproduction. This is a time-consuming approach and costly because the robot remains out of production during the teaching stage.

As the trajectory becomes more and more complex, with many intersecting via points, the operator may experience difficulties while setting up the correct temporal order of the trajectory points. This is the main motivation for the proposal of the neural model described in this paper, since it is highly desirable to have the teaching process with minimal human intervention. In the proposed approach, the responsibility of learning the temporal order of the trajectory is transferred to the neural network and, once training is completed, the stored trajectory can be used for autonomous trajectory planning, as depicted in Fig. 2.

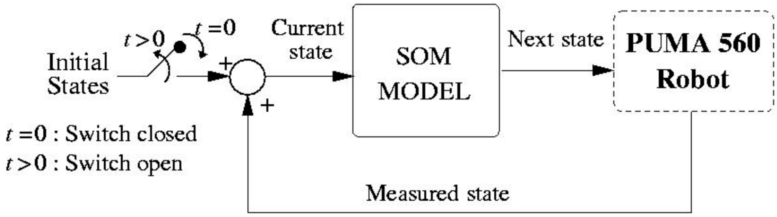


Fig. 2. Autonomous reproduction of a learned trajectory.

Another important issue is the role of the feedback path in Fig. 2, which allows the neural network to work autonomously, performing on a step-by-step basis the reproduction (planning) of the stored trajectory. This is important for safety purposes, since a trajectory only continues to be reproduced is the feedback pathway exists. Thus, if any problem occurs during the execution of the required motion by the robot, such collision with an obstacle or the joints reach their limiting values, the feedback pathway can be interrupted and the reproduction is automatically stopped.

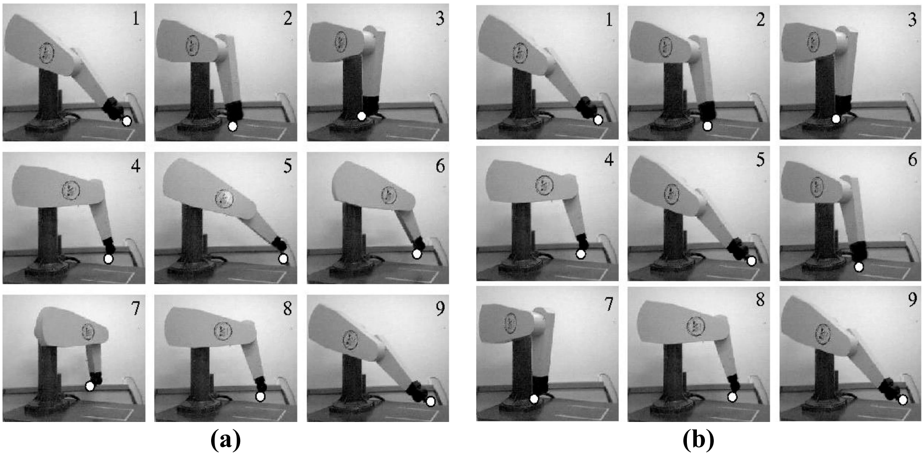
The conventional walk-through method does not possess the feedback pathway. In this case, all the trajectory states are sent to a memory buffer and executed in batch-mode. If any problem occurs, one has to wait for the execution of the whole trajectory in order to take a decision or to turn-off the robot power. Another important property of the VQTAM approach is its greater robustness to noise, as we show next through comparisons with a predictive variant of the lookup table method.

4 Tests with the VQTAM Model

The tests to be presented next evaluate the performance of the proposed method in the tasks of learning and reproduction (planning) of robot trajectories. The following issues will be assessed: accuracy of the retrieved trajectory, influence of the memory parameter (n_θ), influence of the type of neighborhood function, and tolerance of the model to noisy inputs. For this purpose, four trajectories whose pathways approximately describe an eight in 3D Euclidean space were generated by moving the robot through its workspace, each trajectory containing 9 states. This type of trajectory has been used as a benchmark for testing neural learning of robot trajectories because it has a repeated (crossing) via point. Thus, the reproduction of the stored trajectory states in the correct temporal order depends on temporal context information, which is represented by the parameter n_θ.

The proposed approach was implemented in C++ using the PUMA 560 control library QMOTOR/QRTK©, developed by Quality-Real Time Systems, running on a PC under the QNX real-time operating system. More details about the data acquisition processes and the interfacing hardware can be found at http://www.qrts.com and http://www.qnx.com.

The accuracy of the reproduction is evaluated by the *Normalized Root Mean Squared Error* (NRMSE), given by:



(a) **(b)**

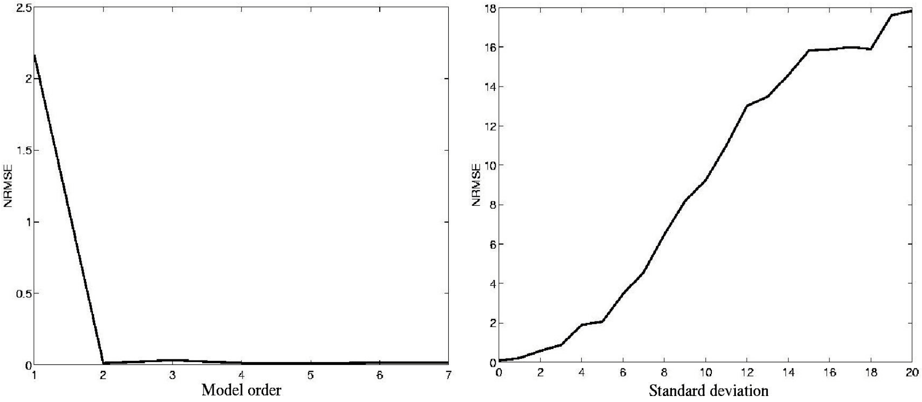
Fig. 3. Influence of the memory on trajectory reproduction: (a) $n_\theta \geq 2$ and (b) $n_\theta = 1$.

$$NRMSE = \sqrt{\frac{1}{N_d . N_s} \sum_{t=1}^{N_d} \sum_{k=1}^{N_s} [\theta_i(k) - \ddot{\hat{\theta}}_i(k)]^2} \tag{14}$$

where N_d is the number of joints of the robot, N_s is the number of states of the trajectory, $\theta_i(k)$ is the desired value of joint *i* at time *k* and $\ddot{\hat{\theta}}_i(k)$ is the retrieved value of joint i at time *k*. The first test demonstrates the ability of the model in retrieving the stored trajectory states in the correct temporal order. We show only the results for one trajectory, since similar results are observed for the other three. A SOM model with 50 neurons was trained with the following parameters: $\eta_0 = 0.9$, $\sigma_0 = 25$, $n_\theta = 2$, $T = 5000$, $N_d = 6$ and $N_s = 9$. A correct reproduction is illustrated in Fig. 3a, where the number at the upper right corner of each subfigure denotes the position in time of that state of the robot arm and the open circle denote the position of the end-effector. An incorrect reproduction occurs if we use $n_\theta = 1$ (no memory!) as shown in Fig. 3b. In this case, the robot is unable to retrieve the whole trajectory, only half of it.

It is interesting to understand why the minimum value of n_θ is 2. This is equivalent to say that, to decide which route to follow, the neural network needs to have information about the current and the last state of the trajectory . This requirement can be easily understood if one notes that, to enter into one half of the trajectory the current state $\theta(t)$ must be the crossing (bifurcation) via point and the last state $\theta(t\text{-}1)$ must be in the other half of the trajectory. This is the minimum memory ``window□□needed to reproduce the trajectory without ambiguity.

Fig. 4a shows the evolution of the NRMSE values as a function of memory parameter. It can be noted that for $n_\theta = 1$, the error is very high and that from $n_\theta \geq 2$ on, the error remains practically the same, confirming the result shown in Fig. 3b.



(a) **(b)**

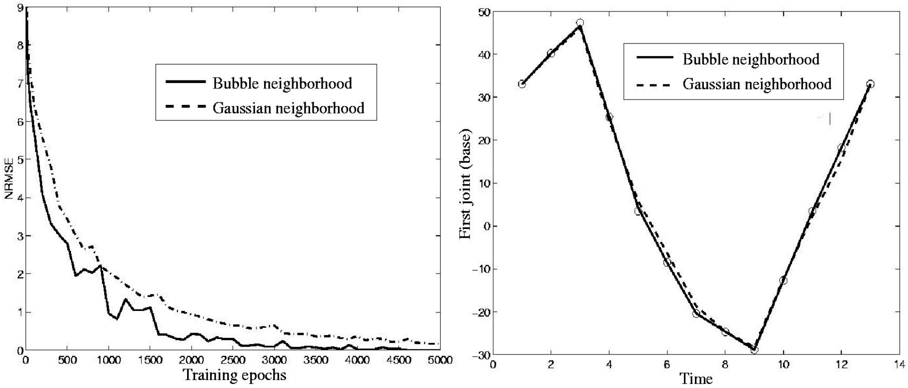
Fig. 4. (a) NRMSE versus memory order. (b) NRMSE versus noise variance.

The next set of tests evaluates the proposed model with respect to the presence of noise in the input vectors and the corresponding effects on the trajectory reproduction. Many tests were performed by adding Gaussian white noise, with zero mean and increasing variance σ^2, to the input vector. For each value of σ^2, we computed the NRMSE associated to the retrieved trajectory states, as shown in Fig. 4b.

One can note that the error increases gradually with the increase of the noise variance, even for very high values of σ^2, which is a direct consequence of the fact of using much more neurons than trajectory states. This occurs due to the topology preserving property of the SOM algorithm, i.e., neurons that are neighbors in the array have close (similar) weight vectors. Thus, a noisy input vector will be mapped onto neurons in the neighborhood of that neuron which would be the winner for the noise-free case. The resulting error is just slightly higher than the noise-free case.

The last test studies the influence of the choice of a neighborhood function $h(i^*, i; t)$ on the numerical accuracy of the reproduction. Two SOM networks were trained for different numbers of training epochs, one with a Gaussian and the other with a rectangular neighborhood function. The results are shown in Fig. 5. From Fig. 5a one can conclude that the rectangular neighborhood always provided lower values for the error. The effect on the retrieved joint angles is illustrated in Fig. 5b, where the retrieved angles of the robot base joint are shown for both types of neighborhood functions. From the exposed, it is recommended the use of the rectangular neighborhood function for two reasons, namely, (1) better accuracy and (2) lower computational cost.

Finally, we discuss the main differences of the model proposed in this paper and the CTH model. The CTH was the first entirely unsupervised neural algorithm applied to trajectory planning and point-to-point control of robotic manipulators, being tested by computer simulations in [2] and by implementation on a real PUMA 560 robot in [5]. The CTH has been applied to predictive modeling of robot trajectories, but it uses different learning mechanisms than those used by the VQTAM method.



(a) **(b)**

Fig. 5. (a) NRMSE versus training epochs. (b) Reproduction accuracy for different types of neighborhood functions.

The main differences are the following:

(i) The CTH uses two sets of weights to learn the temporal order of the states of a trajectory. The first set, comprised by feedforward competitive weights, stores the states, while the second, comprised by lateral Hebbian weights, learns the temporal transitions between consecutive states. The VQTAM needs only feedforward competitive weights to realize the same task.

(ii) The CTH algorithm has 6 training parameters to be determined by experimentation, while the VQTAM uses only two parameters: the learning rate and decay rate of the width of the neighborhood function.

(iii) The VQTAM is more tolerant to noise in the inputs because it benefits from the property of topology preservation of the SOM to produce errors that increase smoothly with the variance of the noise.

5 Conclusions

We introduced a self-supervised neural for prediction and planning of complex robot trajectories. The proposed technique extends Kohonen□s SOM so that it can learn dynamical input-output mappings for trajectory planning purposes. Several tests were carried out using a real PUMA 560 robot, aiming to evaluate the computational properties of the proposed model, such as robustness to noise, influence of the memory order on the model□s performance, and the influence of the type of neighborhood function on the accuracy of the model. The obtained results have shown that the proposed approach performs better the current methods.

Acknowledgments. The authors thank CNPq (DCR:305275/2002-0) and FAPESP (Processes #00/12517-8 and #98/12699-7).

References

[1] Althf□er, K. and Bugmann, G. (1995). Planning and learning goal-directed sequences of robot arm movements, *in* F. Fogelman-Souli□and P. Gallinari (eds), *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, Vol. I, pp. 449□454.

[2] Araj□o, A. F. R. and Barreto, G. A. (2002). A self-organizing context-based approach to tracking of multiple robot trajectories, *Applied Intelligence*, 17(1):99-116.

[3] Barreto, G. A. and Araj□o, A. F. R. (2001). Time in self-organizing maps: An overview of models, *International Journal of ComputerResearch* **10**(2): 139□179.

[4] Barreto, G. A. and Araj□o, A. F. R. (2002). Nonlinear modelling of dynamic systems with the self-organizing map, *Lecture Notes in Computer Science*, 2415:975-980.

[5] Barreto, G. A., , D□cker, C. and Ritter, H. (2002). A distributed robotic control system based on a temporal self-organizing network, *IEEE Transactions on Systems, Man, and Cybernetics-Part C* **32**(4): 347□357.

[6] Bugmann, G., Koay, K. L., Barlow, N., Phillips, M. and Rodney, D. (1998). Stable encoding of robot trajectories using normalised radial basis functions: Application to an autonomous wheelchair, *Proc. 29th Int. Symposium on Robotics (ISR)*, Birmingham, UK, pp. 232□235.

[7] Denham, M. J. and McCabe, S. L. (1995). Robot control using temporal sequence learning, *Proc. World Congress on Neural Networks (WCNN)*, Vol. II, Washington DC, pp. 346□349.

[8] Fu, K., Gonzalez, R. and Lee, C. (1987). *Robotics: Control, Sensing,Vision, and Intelligence*, McGraw-Hill.

[9] Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J. (1992). Neural networks for control systems □A survey, *Automatica* **28**(6): 1083□1112.

[10] Kohonen, T. (1997). *Self-Organizing Maps*, 2nd extended edn,Springer-Verlag, Berlin.

[11] Raibert, M. H. and Horn, B. K. P. (1978). Manipulator control using the configuration space method, *The Industrial Robot* **5**: 69□73.

[12] Ritter, H., Martinetz, T. and Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: An Introduction*, Addison-Wesley, Reading, MA.

[13] Walter, J. and Ritter, H. (1996). Rapid learning with parametrized self-organizing maps, *Neurocomputing* **12**: 131□153.

River Stage Forecasting with Particle Swarm Optimization

Kwokwing Chau

Department of Civil and Structural Engineering, Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. An accurate water stage prediction allows the pertinent authority to issue a forewarning of the impending flood and to implement early evacuation measures when required. Existing methods including rainfall-runoff modeling or statistical techniques entail exogenous input together with a number of assumptions. The use of artificial neural networks has been shown to be a cost-effective technique. But their training, usually with back-propagation algorithm or other gradient algorithms, is featured with certain drawbacks, such as very slow convergence and easily getting stuck in a local minimum. In this paper, a particle swarm optimization model is adopted to train perceptrons. The approach is demonstrated to be feasible and effective by predicting real-time water levels in Shing Mun River of Hong Kong with different lead times on the basis of the upstream gauging stations or stage/time history at the specific station. It is shown from the verification simulations that faster and more accurate results can be acquired.

1 Introduction

Flooding is a type of natural disaster that has been occurring for centuries, but can only be mitigated rather than completely solved. Prediction of river stages becomes an important research topic in hydrologic engineering. An accurate water stage prediction allows the pertinent authority to issue a forewarning of the impending flood and to implement early evacuation measures when required. Currently, environmental prediction and modeling includes a variety of approaches, such as rainfall-runoff modeling or statistical techniques, which entail exogenous input together with a number of assumptions. Conventional numerical modeling addresses the physical problem by solving a highly coupled, non-linear, partial differential equation set which demands huge computing cost and time. However, physical processes affecting flooding occurrence are highly complex and uncertain, and are difficult to be captured in some form of deterministic or statistical model.

During the past decade, the artificial neural networks (ANN), and in particular, the feed forward backward propagation perceptrons, are widely applied in different fields. It is claimed that the multi-layer perceptrons can be trained to approximate and accurately generalize virtually any smooth, measurable function whilst taking no prior assumptions concerning the data distribution. Characteristics, including built-in dynamism in

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1166-1173, 2004.

forecasting, data-error tolerance, and lack of requirements of any exogenous input, render it attractive for use in river stage prediction in hydrologic engineering. Thirumalaiah and Deo [1] depict the use of a conjugate gradient ANN in real-time forecasting of water levels, with verification of untrained data. Liong et al. [2] demonstrate that a feed forward ANN is a highly suitable flow prediction tool yielding a very high degree of water level prediction accuracy in Bangladesh. Chau and Cheng [3] describe the sensitivity of various network characteristics for real-time prediction of water stage with the ANN approach in a river in Hong Kong. Although the back propagation (BP) algorithm is commonly used in recent years to perform the training task, some drawbacks are often encountered in the use of this gradient-based method. They include: the training convergence speed is very slow; it is easily to get stuck in a local minimum. Different algorithms have been proposed in order to resolve these drawbacks, yet the results are still not fully satisfactory [4].

Particle swarm optimization (PSO) is a method for optimizing hard numerical functions based on metaphor of human social interaction [5-6]. Although it is initially developed as a tool for modeling social behavior, the PSO algorithm has been recognized as a computational intelligence technique intimately related to evolutionary algorithms [7-8]. In this paper, a PSO-based neural network approach for river stage prediction is developed by adopting PSO to train multi-layer perceptrons. It is then used to predict real-time water levels in Shing Mun River of Hong Kong with different lead times on the basis of the upstream gauging stations or stage/time history at the specific station.

2 Multi-layer Feed-Forward Perceptron

A multi-layer feed-forward perceptron represents a nonlinear mapping between input vector and output vector through a system of simple interconnected neurons. It is fully connected to every node in the next and previous layer. The output of a neuron is scaled by the connecting weight and fed forward to become an input through a nonlinear activation function to the neurons in the next layer of network. In the course of training, the perceptron is repeatedly presented with the training data. The weights in the network are then adjusted until the errors between the target and the predicted outputs are small enough, or a pre-determined number of epochs is passed. The perceptron is then validated by presenting with an input vector not belonging to the training pairs. The training processes of ANN are usually complex and high dimensional problems. The commonly used gradient-based BP algorithm is a local search method, which easily falls into local optimum point during training.

3 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an optimization paradigm that mimics the ability of human societies to process knowledge. It has roots in two main component methodologies: artificial life (such as bird flocking, fish schooling and swarming); and, evolutionary computation. The key concept of PSO is that potential solutions are flown through hyperspace and are accelerated towards better or more optimum solutions.

3.1 PSO Algorithm

PSO is a populated search method for optimization of continuous nonlinear functions resembling the movement of organisms in a bird flock or fish school. Its paradigm can be implemented in a few lines of computer code and is computationally inexpensive in terms of both memory requirements and speed. It lies somewhere between evolutionary programming and genetic algorithms. As in evolutionary computation paradigms, the concept of fitness is employed and candidate solutions to the problem are termed particles or sometimes individuals. A similarity between PSO and a genetic algorithm is the initialization of the system with a population of random solutions. Instead of employing genetic operators, the evolution of generations of a population of these individuals in such a system is by cooperation and competition among the individuals themselves. Moreover, a randomized velocity is assigned to each potential solution or particle so that it is flown through hyperspace. The adjustment by the particle swarm optimizer is ideally similar to the crossover operation in genetic algorithms whilst the stochastic processes are close to evolutionary programming. The stochastic factors allow thorough search of spaces between regions that are spotted to be relatively good whilst the momentum effect of modifications of the existing velocities leads to exploration of potential regions of the problem domain.

There are five basic principles of swarm intelligence: (1) proximity; (2) quality; (3) diverse response; (4) stability; and, (5) adaptability. The n-dimensional space calculations of the PSO concept are performed over a series of time steps. The population is responding to the quality factors of the previous best individual values and the previous best group values. The allocation of responses between the individual and group values ensures a diversity of response. The principle of stability is adhered to since the population changes its state if and only if the best group value changes. It is adaptive corresponding to the change of the best group value.

In essence, each particle adjusts its flying based on the flying experiences of both itself and its companions. It keeps track of its coordinates in hyperspace which are associated with its previous best fitness solution, and also of its counterpart corresponding to the overall best value acquired thus far by any other particle in the population. Vectors are taken as presentation of particles since most optimization problems are convenient for such variable presentations. The stochastic PSO algorithm has been found to be able to find the global optimum with a large probability and high convergence rate. Hence, it is adopted to train the multi-layer perceptrons, within which matrices learning problems are dealt with.

3.2 Adaptation to Network Training

A three-layered preceptron is chosen for this application case. Here, $W^{[1]}$ and $W^{[2]}$ represent the connection weight matrix between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. When a PSO is employed to train the multi-layer preceptrons, the i-th particle is denoted by

$$W_i = \{W_i^{[1]}, W_i^{[2]}\} \qquad (1)$$

The position representing the previous best fitness value of any particle is recorded and denoted by

$$P_i = \{P_i^{[1]}, P_i^{[2]}\} \tag{2}$$

If, among all the particles in the population, the index of the best particle is represented by the symbol b, then the best matrix is denoted by

$$P_b = \{P_b^{[1]}, P_b^{[2]}\} \tag{3}$$

The velocity of particle i is denoted by

$$V_i = \{V_i^{[1]}, V_i^{[2]}\} \tag{4}$$

If m and n represent the index of matrix row and column, respectively, the manipulation of the particles are as follows

$$V_i^{[j]}(m,n) = V_i^{[j]}(m,n) + r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)] \tag{5}$$
$$+ s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)]$$

and

$$W_i^{[j]} = W_i^{[j]} + V_i^{[j]} \tag{6}$$

where j = 1, 2; m = 1, …, M_j; n= 1, …, N_j; M_j and N_j are the row and column sizes of the matrices W, P, and V; r and s are positive constants; α and β are random numbers in the range from 0 to 1. Equation (5) is employed to compute the new velocity of the particle based on its previous velocity and the distances of its current position from the best experiences both in its own and as a group. In the context of social behavior, the cognition part $r\alpha[P_i^{[j]}(m,n) - W_i^{[j]}(m,n)]$ represents the private thinking of the particle itself whilst the social part $s\beta[P_b^{[j]}(m,n) - W_i^{[j]}(m,n)]$ denotes the collaboration among the particles as a group. Equation (6) then determines the new position according to the new velocity.

The fitness of the i-th particle is expressed in term of an output mean squared error of the neural networks as follows

$$f(W_i) = \frac{1}{S}\sum_{k=1}^{S}\left[\sum_{l=1}^{O}\{t_{kl} - p_{kl}(W_i)\}^2\right] \tag{7}$$

where f is the fitness value, t_{kl} is the target output; p_{kl} is the predicted output based on W_i; S is the number of training set samples; and, O is the number of output neurons.

4 The Study Area

The system is applied to study the tidal dynamics and potential flood hazards in the Shing Mun River network, Hong Kong. Details regarding the location of the Shing Mun River and its tributary nullahs can be found in [9-17]. The existing Shing Mun River has been trained for a length of about 2840m, from the bell-mouth outlet of Lower Shing Mun Dam to Sha Tin Tsuen. The three minor streams, i.e. the Tin Sam, Fo Tan and Siu Lek Yuen nullahs, form tributaries of the extended river. Surface water from an extensive catchment with an area of approximately 5200 ha flows into Sha Tin Hoi via the Shing Mun River. The maximum flow at the river for a 200-year storm is about 1500 m^3/s.

In this study, water levels at Fo Tan is forecasted with a lead time of 1 and 2 days based on the measured daily levels there and at Tin Sam. The data available at these locations pertain to continuous stages from 1999 to 2002, in the form of daily water levels. In total, 1095 pairs of daily levels were available, of which 730 were used for training and 365 were used to validate the network results with the observations. It is ensured that the data series chosen for training and validation comprised both high and low discharge periods of the year and also rapid changes in water stages.

The perceptron has an input layer with one neuron, a hidden layer with three neurons, and output layer with two neurons. The input neuron represents the water stage at the current day whilst the output nodes include the water stages after 1 day and 2 days, respectively. All source data are normalized into the range between 0 and 1, by using the maximum and minimum values of the variable over the whole data sets. In the PSO-based perceptron, the number of population is set to be 40 whilst the maximum and minimum velocity values are 0.25 and -0.25 respectively.

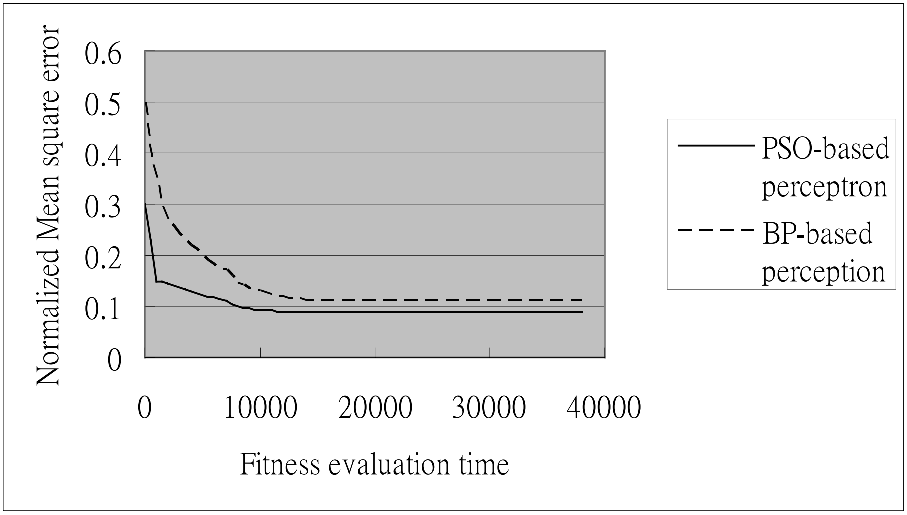


Fig. 1. Relationships between the normalized mean square error and fitness evaluation time during training for PSO-based and BP-based perceptrons

5 Results and Discussions

The PSO-based multi-layer ANN is evaluated along with a commonly used standard BP-based network. In order to furnish a comparable initial state, the training process of the BP-based perceptron commences from the best initial population of the corresponding PSO-based perceptron. Figure 1 shows the relationships between the normalized mean square error and fitness evaluation time during training for PSO-based and BP-based perceptrons whilst Figure 2 shows the 2 day lead time normalized water level prediction by both perceptrons in the validation process. Table 1 and Table 2 show comparisons of the results of network for the two different perceptrons based on data at the same station and at different station, respectively.

The fitness evaluation time here for the PSO-based perceptron is equal to the product of the population with the number of generations. It can be observed that the PSO-based perceptron exhibits much better and faster convergence performance in the training process as well as better prediction ability in the validation process than those by the BP-based perceptron. It can be concluded that the PSO-based perceptron performs better than the BP-based perceptron. Moreover, forecasting at Fo Tan made by using the data collected at the upstream station (Tin Sam) is generally better compared to the data collected at the same location.

Table 1. Results for forecasting at Fo Tan based on data at the same station

Algorithm	Coefficient of correlation			
	Training		Validation	
	1 day ahead	2 days ahead	1 day ahead	2 days ahead
BP-based	0.945	0.913	0.934	0.889
PSO-based	0.974	0.965	0.956	0.944

Table 2. Results for forecasting at Fo Tan based on data at Tin Sam

Algorithm	Coefficient of correlation			
	Training		Validation	
	1 day ahead	2 days ahead	1 day ahead	2 days ahead
BP-based	0.967	0.934	0.954	0.894
PSO-based	0.989	0.982	0.983	0.974

6 Conclusions

This paper presents a PSO-based perceptron approach for real-time prediction of water stage in a river with different lead times on the basis of the upstream gauging stations or stage/time history at the specific station. It is demonstrated that the novel optimization algorithm, which is able to provide model-free estimates in deducing the output from the input, is an appropriate forewarning tool. It is shown from the training and verification simulation that the water stage prediction results are more accurate and are obtained in relatively short computational time, when compared with the commonly used BP-based perceptron. Both the above two factors are important in real-time water resources management. It can be concluded that the PSO-based perceptron performs better than the BP-based perceptron. Moreover, forecasting at Fo Tan made by using the data collected at the upstream station is generally better compared to the data collected at the same location. Future research can be directed towards forecasting river stages by using the more empirical hydrological and rainfall data at the upstream catchment area in order to further extend the lead time of forewarning.

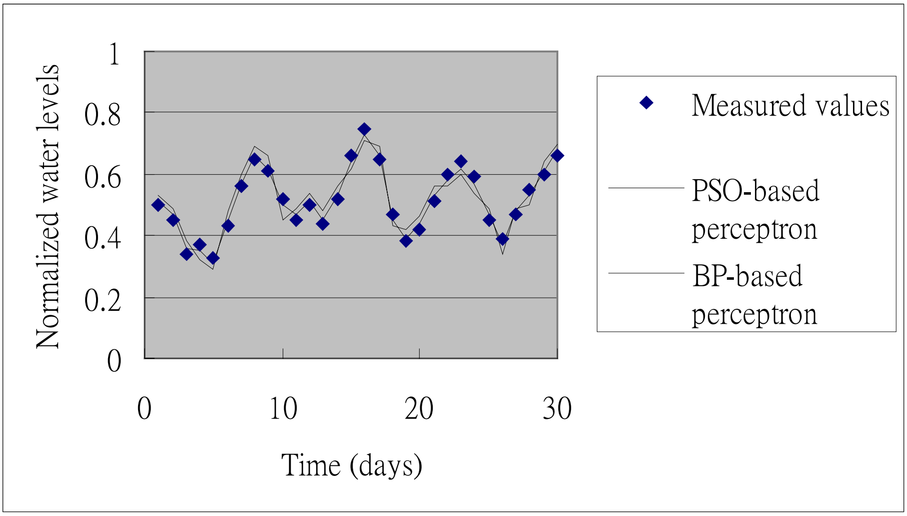


Fig. 2. 2 day lead time water level prediction by both perceptrons in the validation process

Acknowledgement. This research was supported by the Central Research Grant of Hong Kong Polytechnic University (G-T592) and the Internal Competitive Research Grant of Hong Kong Polytechnic University (A-PE26).

References

1. Thirumalaiah, K., Deo, M.C.: River Stage Forecasting Using Artificial Neural Networks. Journal of Hydrologic Engineering, ASCE **3(1)** (1998) 26-32
2. Liong, S.Y., Lim, W.H., Paudyal, G.N.: River Stage Forecasting in Bangladesh: Neural Network Approach. Journal of Computing in Civil Engineering, ASCE **14(1)** (2000) 1-8
3. Chau, K.W., Cheng, C.T.: Real-time Prediction of Water Stage with Artificial Neural Network Approach. Lecture Notes in Artificial Intelligence, **2557** (2002) 715-715
4. Govindaraju, R., Rao, A. (Ed.): Artificial Neural Networks in Hydrology. Kluwer Academic Publishers, Dordrecht (2000)
5. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth (1995) 1942-1948
6. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. Proceedings of the 1997 International Conference on Evolutionary Computation. Indianapolis (1997) 303-308

7. Clerc, M., Kennedy, J.: The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation **6(1)** (2002) 58-73
8. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
9. Chau, K.W.: Manipulation of Numerical Coastal Flow and Water Quality Models. Environmental Modelling & Software **18(2)** (2003) 99-108
10. Chau, K.W.: Intelligent Manipulation of Calibration Parameters in Numerical Modeling. Advances in Environmental Research **8(3-4)** (2004) 467-476
11. Chau, K.W.: A Prototype Knowledge-Based System on Unsteady Open Channel Flow in Water Resources Management. Water International **29(1)** (2004) (in press)
12. Chau, K.W., Chen, W.: A Fifth Generation Numerical Modelling System in Coastal Zone. Applied Mathematical Modelling **25(10)** (2001) 887-900
13. Chau, K.W., Chen, W.: An Example of Expert System on Numerical Modelling System in Coastal Processes. Advances in Engineering Software **32(9)** (2001) 695-703
14. Chau, K.W., Cheng, C., Li, C.W.: Knowledge Management System on Flow and Water Quality Modeling. Expert Systems with Applications **22(4)** (2002) 321-330
15. Chau, K.W., Lee, J.H.W.: Mathematical Modelling of Shing Mun River Network. Advances in Water Resources **14(3)** (1991) 106-112
16. Chau, K.W., Lee, J.H.W.: A Microcomputer Model for Flood Prediction with Applications. Microcomputers in Civil Engineering **6(2)** (1991) 109-121
17. Chau, K.W., Yang, W.W.: Development of an Integrated Expert System for Fluvial Hydrodynamics. Advances in Engineering Software **17(3)** (1993) 165-172

Convergence Analysis of a Neural Network Based on Generalised Compound Gradient Vector

Zaiping Chen[1,2], Xingyun Chen[3], Jianfeng Zhang[2], and Liang Liu[2]

[1] Tianjin University, Tianjin, China
[2] Tianjin University of Technology, Tianjin, China
chenzaiping@eyou.com
[3] Tianjin Medical University, Tianiin China

Abstract. A neural network online training weight update algorithm based on generalised compound gradient vector is introduced. The convergent analysis and proof are made in this paper, which shows that the update of weight values converges to the minimum of the error surface after finite iterations, and the convergent speed of the algorithm is much faster than the BP algorithm. Several simulations have been carried out and the results demonstrate the satisfactory convergent performance and strong robustness obtained using this algorithm for real time control involving uncertainty parameters.

1 Introduction

The ability to learn is a most valuable property of a neural network. Feed-forward neural networks (FNN) have been widely used in various areas. Normally the back propagation algorithm (BP) is used for FNN to update the neural network weight values. Several different BP algorithm improvement schemes have been presented in the literature [1],[2],[3],[4],[6],[13]. To speed up training process, several improving schemes have been investigated in [5],[8],[9],[10]. However most of these improvements are based on the use of heuristic factors to dynamically adapt the learning rate, which only leads to a slight convergence rate improvement [12]. A significant improvement is possible by using various second order approaches such as Newton, conjugate gradient, or the Levenberg-Marquardt (LM) method [7],[10],[11]. The demand for memory to operate with large Jacobians and the necessity to invert large matrices are major disadvantages of the LM algorithm [12]. Because the large number of computations takes significant time, it is difficult to utilize these algorithms in real time control systems.

A new neural networks learning scheme based on a generalised compound gradient vector is introduced and the convergent analysis and proof are carried out in this paper. The convergent analysis indicates that this algorithm can assure that the network weight update converges to the minimum of the error surface after finite iterations, and the convergent speed can outperform that of the standard BP algorithm.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1174-1183, 2004.

2 Analysis of BP Algorithm Convergent Speed

The standard BP weight updating formula (without momentum) can be written as below:

$$w(k+1) = w(k) - \eta \frac{\partial E(k)}{\partial w(k)} \tag{1}$$

The incremental form of equation (1) can be written as follows:

$$\Delta w(k) = -\eta \frac{\partial E(k)}{\partial w(k)} \tag{2}$$

In the same way, the weight updating formula of the (k-1)th and (k-2)th steps can be written as follows:

$$\Delta w(k-1) = -\eta \frac{\partial E(k-1)}{\partial w(k-1)} \tag{3}$$

$$\Delta w(k-2) = -\eta \frac{\partial E(k-2)}{\partial w(k-2)} \tag{4}$$

Considering the orthogonal relationship between the search directions of two successive iterations in the gradient descent algorithm, we can get the equation as the following[14]:

$$[-\nabla E(w_k)] \cdot [-\nabla E(w_{k-1})] = 0 \tag{5}$$

where $\nabla E(w_k)$ is the kth iteration of the error function in the neural network gradient descent weight value space. Therefore, the Fig.1 shows the gradient vectors of update weight process of the standard BP algorithm.

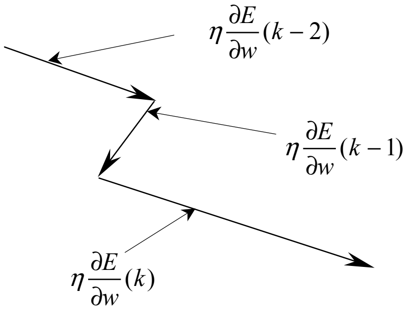

$\eta \frac{\partial E}{\partial w}(k-2)$
$\eta \frac{\partial E}{\partial w}(k-1)$
$\eta \frac{\partial E}{\partial w}(k)$

Fig. 1. Diagram of the successive gradient descent search vectors of the standard BP algorithm.

By means of this gradient descent standard BP algorithm, it is evident from Fig.1 that the search process to the minimum is always tortuous with right angle, which is the main reason why the convergent speed is affected. In order to improve BP algorithm and increase the learning convergent speed, the improved algorithm for overcoming the orthogonal gradient descent search should be investigated.

3 A New Architecture of the Weight Updating Algorithm

According to the basic concept mentioned above, now the improved weight updating formula can be written as following formation:

$$\Delta w(k) = -\eta \frac{\partial \widetilde{E}(k)}{\partial w(k)} \tag{6}$$

where $\eta \frac{\partial \widetilde{E}(k)}{\partial w(k)}$ is called generalised compound gradient vector(GCGV) and can be defined as the following :

$$\eta \frac{\partial \widetilde{E}(k)}{\partial w(k)} = \eta \frac{\partial E(k)}{\partial w(k)} + \eta \frac{\partial E(k-1)}{\partial w(k-1)} + \cdots\cdots + \eta \frac{\partial E(k-n)}{\partial w(k-n)} \tag{7}$$

where $n = 1,2...m$

According to the orthogonal relation of gradient descent search direction, the generalised compound gradient vector of equation (7) can be described in Fig.2.

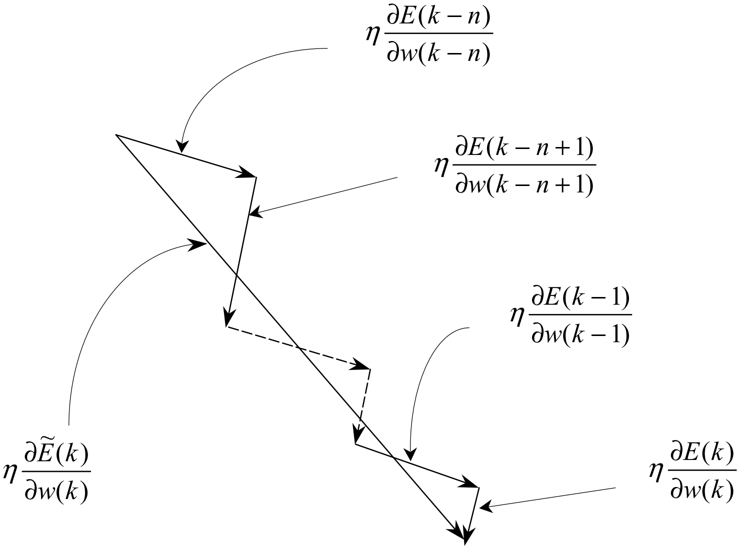

$\eta \frac{\partial E(k-n)}{\partial w(k-n)}$
$\eta \frac{\partial E(k-n+1)}{\partial w(k-n+1)}$
$\eta \frac{\partial E(k-1)}{\partial w(k-1)}$
$\eta \frac{\partial \widetilde{E}(k)}{\partial w(k)}$
$\eta \frac{\partial E(k)}{\partial w(k)}$

Fig. 2. Diagram of generalised compound gradient vector

It is evident that the historical information of gradient descend search has a remarkable influence on weight update processes, which can be controlled by changing the architecture of GCGV. In general, the successive generalised compound gradient vectors are not mutually orthogonal.

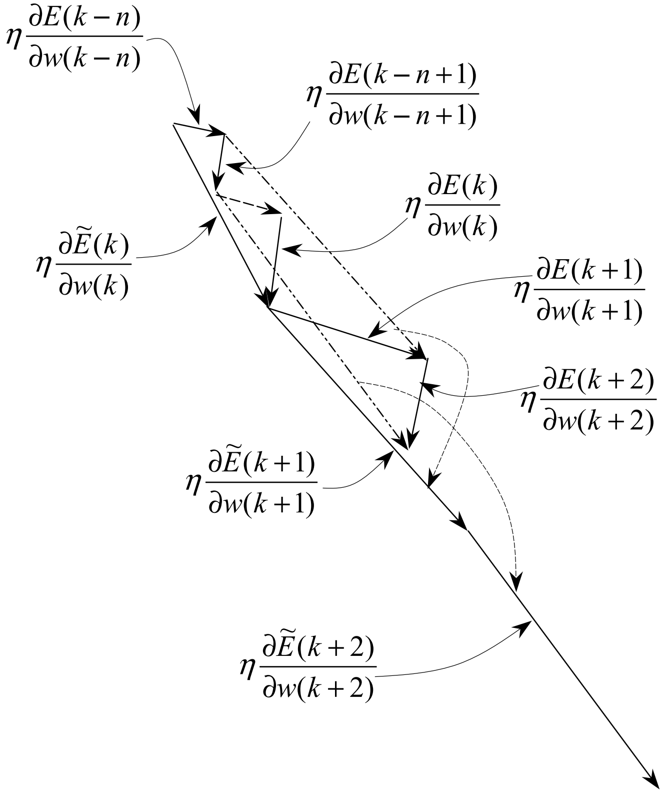

$\eta \frac{\partial E(k-n)}{\partial w(k-n)}$
$\eta \frac{\partial E(k-n+1)}{\partial w(k-n+1)}$
$\eta \frac{\partial E(k)}{\partial w(k)}$
$\eta \frac{\partial \tilde{E}(k)}{\partial w(k)}$
$\eta \frac{\partial E(k+1)}{\partial w(k+1)}$
$\eta \frac{\partial E(k+2)}{\partial w(k+2)}$
$\eta \frac{\partial \tilde{E}(k+1)}{\partial w(k+1)}$
$\eta \frac{\partial \tilde{E}(k+2)}{\partial w(k+2)}$

Fig. 3. Diagram of successive generalised compound gradient vector.

The relationship between the successive generalised compound gradient vectors is given in Fig. 3. Therefore, it is not difficult to understand that this improved algorithm is faster than standard BP algorithm in learning and convergent rate by means of this search path to locate error minimum.

4 Convergent Analysis of Generalised Compound Gradient Vector Algorithm

It is evident from Fig.2 that neural network weight updating algorithm fully extract the known information. Therefore, the suitable selection of the formed steps of generalised compound gradient vector usually can achieve more correct and effective gradient descend directions.

Let

$$a_k = \frac{\partial E(k)}{\partial w(k)} \tag{8}$$

$$a_{k-1} = \frac{\partial E(k-1)}{\partial w(k-1)} \tag{9}$$

$$a_{k-2} = \frac{\partial E(k-2)}{\partial w(k-2)} \tag{10}$$

$$a_{k-n} = \frac{\partial E(k-n)}{\partial w(k-n)} \tag{11}$$

$$a_{k-n-1} = \frac{\partial E(k-n-1)}{\partial w(k-n-1)} \tag{12}$$

According to BP algorithm,

$$\Delta w(k) = -\eta \frac{\partial E(k)}{\partial w(k)} \tag{13}$$

It can be known that weight learning update converges to the minimum of the error surface after finite iterations when η is chosen to meet the relation $\sup\|Q(w)\| \le \eta^{-1} < \infty$ in some bounded region where the relation $E(w) \le E(w_0)$ [11], where $Q(w)$ denotes the cost function with respect to the Hessian matrix in weight space, and w_0 denotes the initial weight vector.

However, the computational requirements of the Hessian for FNNs with several hundred weights make its use impractical. Thus, the learning rate factor is usually chosen according to the relation $0< \eta < 1$. The learning rate satisfied with this condition can finally make the weight learning updating converge to the minimum of the error surface after a finite iterations.

Therefore, when the times of learning iteration is much enough, there is

$$\left|\eta\frac{\partial E(k)}{\partial w(k)}-\eta\frac{\partial E(k-1)}{\partial w(k-1)}\right|<\frac{\varepsilon^*}{n} \tag{14}$$

where $\varepsilon^* > 0$ and $n = 1,2 \ldots m$.

The equation (14) can be arranged as:

$$\left|\frac{\partial E(k)}{\partial w(k)}-\frac{\partial E(k-1)}{\partial w(k-1)}\right|<\frac{\varepsilon}{n} \tag{15}$$

where $\varepsilon = \dfrac{\varepsilon^*}{\eta}$.

According to Eq (8)-(12), yields

$$\left|a_k - a_{k-1}\right| < \frac{\varepsilon}{n} \tag{16}$$

$$\left|a_{k-1} - a_{k-2}\right| < \frac{\varepsilon}{n} \tag{17}$$

$$\vdots$$

$$\left|a_{k-n} - a_{k-n-1}\right| < \frac{\varepsilon}{n} \tag{18}$$

Considering

$$\begin{aligned}&\left|a_k + a_{k-1} + \cdots + a_{k-n} - (a_{k-1} + a_{k-2} + \cdots + a_{k-n-1})\right| \le \\ &\left|a_k - a_{k-1}\right| + \left|a_{k-1} - a_{k-2}\right| + \cdots + \left|a_{k-n} - a_{k-n-1}\right| < \varepsilon\end{aligned} \tag{19}$$

and

$$\frac{\partial \widetilde{E}(k)}{\partial w(k)} = a_k + a_{k-1} + \cdots + a_{k-n} \tag{20}$$

$$\frac{\partial \widetilde{E}(k-1)}{\partial w(k-1)} = a_{k-1} + a_{k-2} + \cdots + a_{k-n-1} \tag{21}$$

From equation (19), (20) and (21), we can get:

$$\left|\frac{\partial \tilde{E}(k)}{\partial w(k)} - \frac{\partial \tilde{E}(k-1)}{\partial w(k-1)}\right| < \varepsilon \tag{22}$$

It is evident from equation (22) that the weight learning update must converge to stable state when η is chosen to meet the convergent condition and the times of learning iteration is much enough. Thus, it ensures that the gradient descent speed search converges to the minimum of the error surface.

By analysing the generalised compound gradient vector shown in Fig.2, it is evident that this algorithm, by means of generalised compound gradient vector, avoids the tortuous search route with right angles in standard BP algorithm during gradient descent search processes. Thus, the efficiency is improved and the convergent speed of this algorithm is obviously superior to that of standard BP algorithm. Besides, this algorithm is quite suitable to the requirement of real time control because only the known information is applied, not involving complicated computation.

5 Algorithm Simulation

Simulation results of GCGV algorithm are given here in order to verify the new algorithmic convergent speed when the plant parameters vary. The inverter dynamic controller based on neural networks is utilised, while feed-forward controller adopts conventional controller. The simulation model of the plant is given as follows:

$$y_k = 1.9979y_{k-1} - 0.998y_{k-2} + 0.502u_k + 0.0005u_{k-1} - 0.0487u_{k-2} \tag{23}$$

In this model there are two poles at (-0.1001+*i*0.9955, -0.1001-*i*0.9955). The three-layer architecture of neural network in the system is used as in Fig.4. There are two neurons in input layer, six neurons in hidden layer and one in output layer. In GCGV algorithm, three step compound vectors and two momentum items are used.

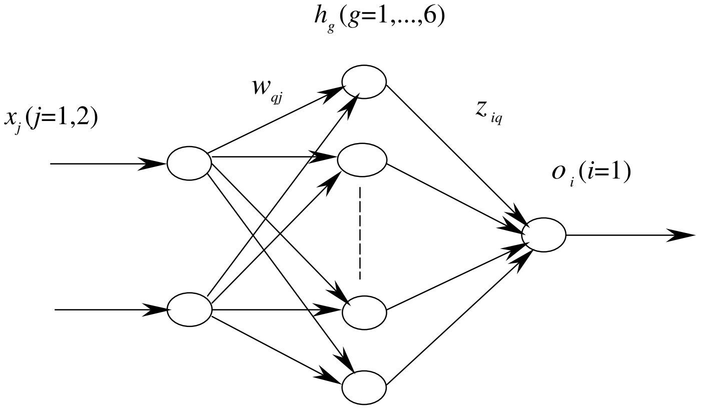


Fig. 4. Diagram of neural network construction

In the simulation process, disturbance of the plant parameter occurs at 50th iteration, which lasts 10 time iterations, and one of the plant parameters changes from 1.9979 to 1.97 at 350th iteration.

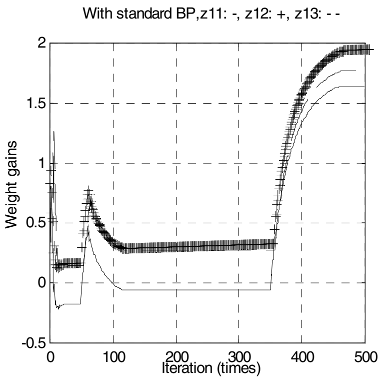


Fig. 5. Some output layer weight learning processes with standard BP algorithm

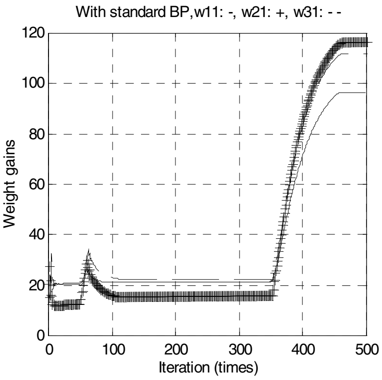


Fig. 6. Some hidden layer weight learning processes with standard BP algorithm

Under the disturbance and the variations of parameter, the simulations with GCGV algorithm and standard BP algorithm are given in Fig.5-Fig.10 respectively. Due to the large range changes of plant pole positions, the learning of the standard BP algorithm is not yet complete until 450th iteration about 100 time iterations lasted, which can be seen in Fig.5 and Fig.6. The online learning procedures of GCGV algorithm are shown in Fig. 7 and Fig. 8. System step responses under the disturbance and the variations of parameter with BP and GCGV algorithms are given in Fig.9 and Fig.10 respectively. The contrast of GCGV neural networks algorithm and standard BP algorithm, the convergent speed of GCGV neural networks online learning algorithm is much faster than that of the standard BP algorithm, and the learning procedure can be finished only lasting about 50 time iterations using GCGV algorithm.

6 Conclusions

The GCGV neural network scheme is proposed and the convergent analysis and proof of the scheme are carried out. The GCGV neural network scheme overcomes the drawbacks of using heuristic factors and the large computation demands of other schemes. Several simulations have been carried out and the results verify the satisfactory convergent performance and strong robustness obtained using this algorithm for real time control involving uncertainty parameters.

Acknowledgements. The authors acknowledge with thanks the financial support by Tianjin Nature Science Foundation, Tianjin Education Committee Key Discipline Grant and Tianjin University of Technology.

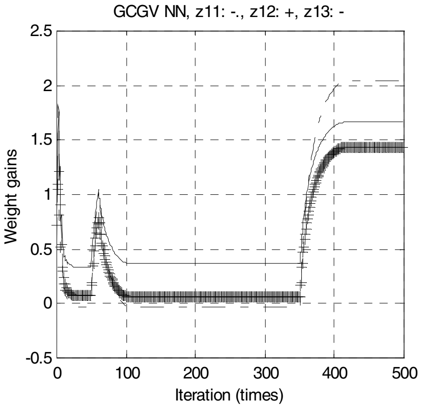


Fig. 7. Some output layer weight learning processes with standard GCGV algorithm

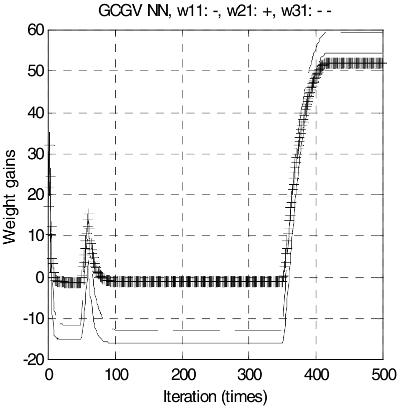


Fig. 8. Some output layer weight learning processes with standard GCGV algorithm

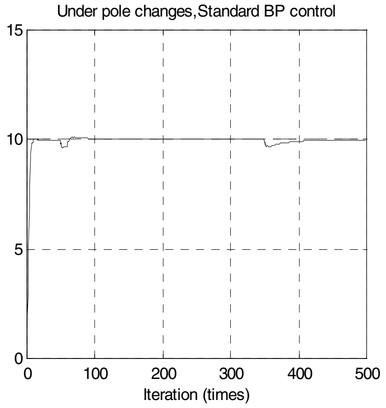


Fig. 9. System step response with standard BP algorithm

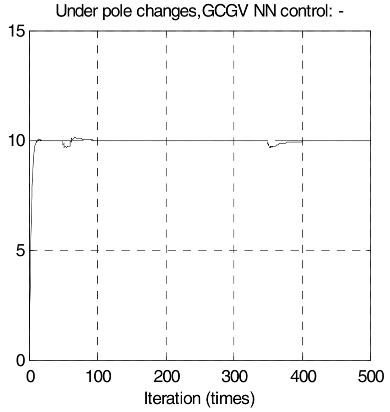


Fig. 10. System step response with GCGV algorithm

References

1. Kuan Chung-Ming, Hornik Kurt: Convergence of Learning Algorithms with Constant Learning Rates. IEEE Transactions on Neural Networks, vol.2 (5, 1991) 484-489
2. Ngolediage J.E., Naguib R.N.G, Dlay S.S.: Fast Back-Propagation for Supervised Learning. Proceedings of 1993 Internatioanl Joint Conference on Neural Networks, (1993) 2591-2594
3. Maugoulas G.D., Vrahatis M.N., Androulakis G.S.: Effective Backpropagation Training with variable stepsize, Neural Networks, (1,1997) 69-82
4. Van der Smagt P.P.: Minimisation methods for Training Feedforward Neural networks, Neural Networks, (1, 1994) 1-11

5. Van Ooyen A., Nienhuis B.: Improving the convergence of the Back-Propagation Algorithm, Neural Networks, (3,1992) 465-471
6. Zhou G. Si J.: Advanced Neural Networks Training Algorithm with Reduced Complexity based on Jacobian Deficiency, IEEE Transactions on Neural Networks, (3,1998) 448-453
7. Hagan M.T., Menhaj M.B.: Training feed-forward Neural Networks with the Marquardt Algorithm, IEEE Transactions on Neural Networks, (6, 1994) 989-993
8. Samad T.: Back-propagation Improvements based Heuristic Arguments, Proceedings of International joint Conference on Neural Networks, (1990) 565-568
9. Bello M. G.: Enhanced Training Algorithms, and Integrated Training/Architecture Selection for Multilayer Perceptron Networks, IEEE Transactions on Neural networks, (6,1992) 864-875
10. Shah S. Palmieri F.: MEKA-A Fast, Local Algorithm for Training Feed-forward Neural Networks, Proceedings of international Joint Conference on neural Networks, (1990) 41-46
11. Parisi R., Di Claudio E. D., Orlandi G., Rao B. D.: A generalized Learning Paradigm Exploiting the Structure of Feed-forward Neural Networks, IEEE Transactions on Neural networks, (6,1996) 1450-1459
12. Wilamowski Bogdan M.,Iqlikci Serdar, Kaynak Okyay, Onder Efe M.: An Algorithm for Fast Convergence in Training Neural Networks, IEEE Proceedings of International Joint Conference on Neural Networks, (2001) 1778-1782
13. Zaiping Chen, Jun LI, Hui Zhao, Qiang Gao, Youjun Yue, Zhenlin Xu: Online Training of Neural Network Control for Electric Motor Drives. The Proceedings of IEEE Conference on Systems, Man and Cybernetics 2002
14. Xu Lina: Neural Networks Control. Harbin Industrial University Press, Harbin (1999) 123-124

Locating Oil Spill in SAR Images Using Wavelets and Region Growing

R□gia T.S. Araj□o, F□tima N.S. de Medeiros, Rodrigo C.S. Costa,
R□gis C.P. Marques, Rafael B. Moreira, and Jilseph L. Silva

UFC/DETI □C. P. 6007, Cep 60455-760 Fortaleza, CE, Brazil
{regia,fsombra,rodcosta,regismarques,rafaelbarbosa,jilseph}@deti.ufc.br

Abstract. This paper presents an algorithm for spots detection in Synthetic Aperture Radar (SAR) images that can be used to support environmental remote monitoring. Monitoring areas with high frequency of oil spillage by accidental or illegal oil discharges can prevent marine damage spreading. But the presence of speckle noise in SAR images limits the visual interpretation of scenes because it obscures the content. Thus, to get reliable data interpretation and quantitative spots measurements, it is recommended to applying speckle filtering schemes. We propose an algorithm to locate dark areas in the sea that are candidate to be oil slicks by combining region growing approach and multiscale analysis. The multiscale analysis employed by the undecimated wavelet smooths the speckle noise in SAR images while enhances edges. The proposed algorithm provides a better segmentation result that is achieved by a modified region growing approach. The algorithms were tested in real SAR images with oil spillages in the sea.

1 Introduction

SAR data have been increasing used for monitoring and managing environmental changes in the Earth□s surface. For marine applications the SAR systems have improved the possibilities for oil spills detection, that seriously affect the marine ecosystem, allowing a more rigorous and cost effective monitoring. Spillage of oil in coastal waters can be a catastrophic event. The potential damage to the environment and economy of the area at stake requires that agencies be prepared to rapidly detect, monitor, and clean up any large spill [1].

Oil spill detection in the sea is determined by the contrast between oil spectral radiance and water radiance around the oil. The presence of an oil film on the sea surface damps the small waves due to the increased viscosity of the top layer and drastically reduces the measured backscattering energy, resulting in darker areas in SAR imagery [2].

An analysis of remotely sensed imagery of the Exxon Valdez oil spill in Alaska, on 24 March 1989 showed that the remotely-sensed images provide precise information of the geographic position and extension of the oil spill [3]. Automatic identification of oil spills in SAR images is a very complex task because similar images of oil slicks frequently occur, particularly in low-wind conditions [4]. A careful interpretation is

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1184-1193, 2004.

required then. In general, the visual operator determines if a dark object is an oil spill or a similar image.

Studies have been carried out to improve automatic and semi-automatic methods to detect oil spills using SAR images. Solberg *et al.* [4] developed a semi-automatic system for oil spill detection, in which objects with a high probability of being oil slicks are automatically identified in SAR images. This methodology for oil spills detection followed these steps: 1) selection of an area of interest in the image; 2) filtering of the noisy image; 3) segmentation of the image area; 4) extraction and analysis of the geometrical shape of the slicks; 5) features extraction from the isolated area; 6) classification of the dark areas into oil slick or look-alike, based on the features.

In this paper we have developed algorithms related to the steps 2, 3, 4 and 5 and further developments will include the step 6.

Liu *et al.* [5] proposed algorithms to detect and track mesoscale oceanic features employing multiscale wavelet analysis using the 2-D Gaussian wavelet transform to track oil slicks, eddies, fronts, whirlwinds and icebergs. The authors concluded that the wavelet analysis can provide a more cost-effective monitoring program to keep track of changes in important elements of the coastal watch system.

A neural network approach for oil spills detection in European Remote Sensing Satellite (ERS-SAR) imagery was explored by Del Frate et al. [6]. They proposed an algorithm to classify oil slicks based on a set of geometrical features extracted from real oil slicks images and similar ones. The features extracted from dark objects were the perimeter, complexity, area, spreading, standard deviation of the object and of the background, maximum and mean contrast and gradient standard deviation [6]. The input of the network consisted of a set of features providing information about an oil spill candidate and the output concerned the probability for the candidate to be a real oil spill. The authors concluded that the introduction of physical characteristics related to atmospheric conditions such as wind speed and water temperature would improve the algorithm results.

In this paper we propose an algorithm to detect spills in Synthetic Aperture Radar (SAR) images that can be used to support environmental remote monitoring. The methodology combines region growing approach and multiscale analysis to locate dark areas in the sea. To improve the segmentation task we apply the Min/Max algorithm [7], as a post processing task, which eliminates the remnant noise in the segmented image. Section 2 presents the background of the methodology used, pointing out the filtering and segmentation techniques applied in this article. In Section 3 we introduce the proposed algorithm. Section 4 presents the geometrical features computation of the slicks and the analysis of the results. Section 5 contains the experimental results and the last section concludes with discussions and remarks.

2 Background

In this section, we present the background of the methodology used in this paper for automatic detection of oil spills using SAR images that includes image filtering and segmentation.

2.1 Wavelet Filtering

The wavelet transform is an efficient band-pass filter, which can be used to separate various scales of processes and show their relative phase/location information [2].

The continuous-time version of the wavelet transform applied to a function $f(r)$ in terms of the complex wavelet function, $\psi(r)$, is expressed as follows [8]:

$$W_f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(r)\psi^*(\frac{r-b}{a})dr \ . \tag{1}$$

$W_f(a,b)$ corresponds to the wavelet transform coefficients, $\psi(r)$ is the wavelet basis or mother wavelet, a is the scale parameter and b indicates the location parameter, i. e., the basis function $\psi(r)$ is dilated by a factor a, and shifted by b. This function must satisfy the admissibility condition, but it is otherwise subject to choice within certain limits [5]. The superscript (*) indicates complex conjugate.

In the literature there are some papers that approach noise filtering by applying hard or soft thresholding to wavelet coefficients. Images often are degraded in the process of capture, transmission or storage requiring noise reduction to provide a better image quality. In many situations the distortion process can be modeled as an additive Gaussian noise.

SAR images are generally affected by speckle that is a kind of multiplicative noise. Speckle phenomenon results from the coherent radiation which appears as a granular signal-dependent noise, whose effect is to degrade the image quality. In [10] Medeiros et al. proposed a filtering algorithm that combined orthogonal wavelet transform and adaptive windowing. This filter was able to reduce speckle in SAR images and to preserve edges.

In this paper we achieve the noise filtering by decomposing the noisy image using the algorithm *à trous* [8]. This algorithm is an undecimated discrete wavelet transform, which means that each scale of the decomposed image has the same dimension as the original one.

The first step of the algorithm consists in obtaining the scalar product of a signal $f(x)$ with a scaling function $\varphi(x)$ which corresponds to a low-pass filter [9] generating the sequence of the approximation coefficients, c_0. We have used the triangle scaling function.

The distance between samples of the sequences increases by a factor 2 from scale j-1 (j>0) to the next one (j). The gradual increasing of the smoothing effect in each scale is a consequence of the scaling function dilation by inserting zeroes, which originates the algorithm□s name, *à trous*. The subsequent smoothed sequences $c_j(k)$ are given by:

$$c_j(k) = \sum_l h(l)c_{j-1}(k+2^{j-1}l) \ . \tag{2}$$

The filter h is given by the equation:

$$h(l) = \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{x}{2}\right), \phi(x-l) \right\rangle \ . \tag{3}$$

In the next step of the algorithm the wavelets coefficients, $w_j(k)$, are calculated by:

$$w_j(k) = c_{j-1}(k) - c_j(k) \ . \tag{4}$$

The image is reconstructed by adding the wavelets coefficients of all scales and the last scale of the approximation coefficients, c_n, using the equation:

$$c_0(k) = c_n(k) + \sum_{j=0}^{n} w_j(k) \ . \tag{5}$$

The most relevant features of the signal appear in some scales although being gradually smoothed. It is worth noting that the performance of the filtering algorithms based on the wavelet transform depends on the coefficients thresholding method. In this work we adopted a scheme inspired in [13] to achieve the wavelets coefficients filtering.

2.2 Segmentation Using Region Growing

Segmentation of SAR images is an important step in applications such as visualization, quantitative analysis and image-guided remote monitoring. Image segmentation is based on differences in characteristics between regions and/or specific features found at image borders. Numerous segmentation methods have been developed in the literature. Traditional methods, such as pixel-based clustering, region growing, and edge detection require additional pre-processing and post-processing as well as a considerable amount of expert intervention and/or a prior information of an object of interest.

The objective of segmentation is the partition of an image into regions based on the numbers of objects in the scene, using one or more rule of similarity [11]. A typical similarity rule is the gray level. Other region properties such as colour and texture generally require a more elaborate feature extraction over local neighborhoods. Region-based segmentation schemes, such as histogram thresholding and region growing algorithms try to define regions by their content. The simplest region growing method is the pixel aggregation, where the regions start growing around the seed pixels if the neighbors have same characteristic of color, gray level or texture [11]. The region stops growing around the seed when the pixels in the neighborhood no longer obey the similarity rule.

3 The Proposed Algorithm

The proposed algorithm achieves the noise filtering by applying an undecimated wavelet transform to improve the quality of SAR images. After filtering these images the region growing process starts.

Fig. 1 displays the block diagram of the proposed algorithm. The filtering module is based on the algorithm proposed in [12] and adapted by Marques et al. [13]. It identifies relevant features such as edges in noisy signals by correlating scales of the non orthogonal wavelet decomposition. The peaks in the correlation result lead to edges and noise information in SAR images. The variance around these peaks is used to differ signal peaks from noise ones. The method smoothes the signal peaks applying a Gaussian filter and reduces the peaks due to noise employing a simple hard thresholding. The *à trous* algorithm was implemented considering the triangle function as the scaling function.

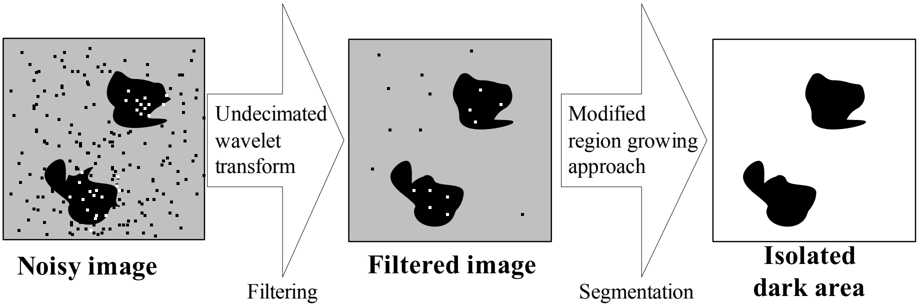


Fig. 1. Block diagram of the proposed algorithm

The segmentation module includes a modified version of the region growing approach. At first, it performs an automatic search of the seed pixels in the smoothed histogram. The reason for smoothing the histogram of the SAR image using a Gaussian mask, consists in reducing the number of seed candidates. The smoothed histogram is decomposed by the *à trous* algorithm and the first scale of decomposition is correlated with the smoothed one in order to locate the maximum and minimum peaks. The seeds are chosen according to the magnitude of the peaks in the correlated signal. When a peak in the correlation result is higher than the coefficient of the wavelet decomposition it is preserved and it points towards a seed. The region grows around seed pixels using the following rule of aggregation:

1. The absolute difference in gray level between the seed pixel and the candidate to be incorporated to the region not exceed 15% of the difference between the minimum and maximum gray levels in the entire image (255 in this case).
2. The pixel candidate is 4-connected to the seed pixel.
3. If one pixel belongs to 2 regions (for example oil or water), the pixel is arbitrarily assigned to the region with higher gray level.
4. The region stops growing if the first or the second rule is no more obeyed.

4 Computation and Analysis of the Geometrical Features of the Slicks Shapes

After the segmentation process, the binary image enhancement is provided by the Min/Max method [7], prior to calculating the geometrical measures. This scheme switches between removing noise and maintaining essential properties of the image. It means that the correct flow is based on the neighborhood characteristics.

The geometrical pattern of slicks is an important characteristic to be measured from SAR images [14]. Furthermore, it is possible to track oil spills by combining geometrical and mesoscale features. To compute geometrical features of oil spill patterns, we select some characteristics that concern an oil spill signature. These features are relevant to evaluate slicks dimension, location, destination and shape, etc.

The uncalibrated area (A) and perimeter (P) in pixels indicate the dimension of an oil spill. The area of an oil spill candidate is the number of inner pixels of the dark region. The perimeter (pixels) is calculated by counting the number of edge points of the spot. Derived from the area and perimeter measures, the complexity (Comp.) takes a small numerical value for region with simple geometry and larger values for complex geometrical regions. The area to perimeter ratio is also considered. The center of mass (C.M.) can be used to specify the location of an object [15]. Derived from the center of mass, the maximum (D_{max}) and minimum (D_{min}) distances from the boundary points to the center of mass are useful to determine qualitatively the elongation degree of the oil spill. The spreading (Spr) is derived from the principal component analysis [6]. This measure will be low for long and thin objects and high for objects closer to a circular shape. Other important features are: maximum (G_{max}) and mean values (G_{me}) of the border gradient; object (Osd) and background (Bsd) standard deviation; standard deviation of the gradient values of the object border (Gsd); maximum (C_{max}) and mean contrast (C_{me}). The formulas of these measures are shown in Appendix A.

5 Experimental Results

We tested our algorithm on real SAR images, but only report results for Fig. 2a. It is a real SAR image of an oil spill accident. It is provided in the site http://earth.esa.int/ew/oil_slicks/north_sea_96/. The scene is in the North Sea and it has been acquired from the ERS-2 satellite on July 18 1996. Figs. 2b and 2c show the results of the proposed algorithm applied to the smoothed SAR image using the undecimated wavelet decomposition and applied to the noisy image, respectively. Fig. 2d illustrates the use of the Min/Max scheme to enhance the result presented in Fig. 2c. Fig. 2e displays the original image segmented by Liu□s algorithm [5].

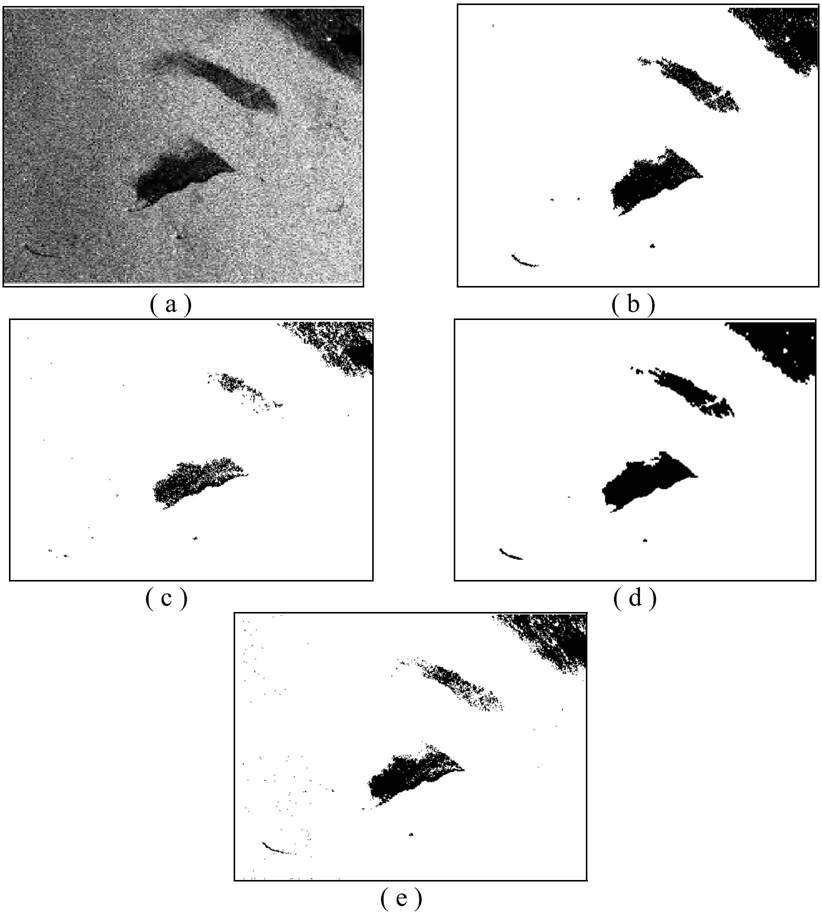


Fig. 2. (a) The original image (b) the proposed algorithm applied to the smoothed image (c) the proposed algorithm applied to the original image (d) the post-processing result obtained after applying the proposed algorithm to the smoothed original image and (e) the Liu□s algorithm result

Table 1 presents the geometrical features extracted from the central spot in the images presented in Figs.: 2b (Alg.01 - the smoothed SAR image segmented by the proposed scheme), 2c (Alg. 02□the original SAR image processed by the proposed algorithm), 2d (Alg. 03□the image in Fig.2b post-processed by the Min/Max scheme) and 2e (Alg.04□the original SAR image processed by Liu□s algorithm). The columns of Table 1 display the results for each algorithm and the lines information represents the geometrical measures that characterize the oil slick candidate. The segmentation methods used to isolate the spots can be assessed by evaluating the differences between these calculated measures.

According to Table 1 it can be observed that the post processing Min/Max algorithm distorted the area of the spot in the center of Fig. 2e increasing it less than 1%. In fact, it is an irrelevant distortion considering the complexity of the problem.

The perimeter decreased considerably, due to the noise filtering smoothing effect. The other measures were slightly modified.

Table 1. Signature measures of a real oil spill

Feature	Alg. 01	Alg. 02	Alg. 03	Alg. 04
A	57240	33414	58574	45907
P	9613	16687	2592	10345
C.M.	(216,717)	(225,723)	(213,721)	(221,715)
D_{max}	762,15	758,55	766,61	857,01
D_{min}	0	1	1	0
Spr.	35,49	37,92	35,10	36,07
Comp.	11,33	25,75	3,02	13,62
A/P	5,95	2,00	22,59	4,43
G_{max}	20,79	14,91	19,24	18,23
G_{me}	11,69	9,08	12,41	11,26
Osd	14,36	11,57	14,96	13,74
Bsd	17,33	17,43	17,34	17,37
Gsd	12,84	9,99	13,43	12,22
C_{max}	21,36	21,27	21,36	21,32
C_{me}	20,16	20,71	20,03	20,39

6 Concluding Remarks

In this paper a novel method for automatic multiscale segmentation was presented to support environmental remote monitoring using SAR images. To demonstrate the robustness of this method, it has been applied to real SAR images. The results showed that the proposed algorithm produced the best segmentation of the dark areas including the oil spills. The results provided by the Min/Max algorithm combined to the proposed algorithm were quite satisfactory.

We can observe that the smoothing effect employed by the *à trous* algorithm led to a better performance of the segmentation algorithm.

Acknowledgements. The authors are grateful to CNPq and FUNCAP for financial support.

References

1. Marghany, M.: Radarsat Automatic Algorithms for Detecting Coastal Oil Spill Pollution. Asian Journal of Geoinformatics, Vol.3 (2) (2001) 191- 196
2. Trivero, P., Fiscella, B., Gomez, F., Pavese, P.: SAR Detection and Characterization of Sea surface Slicks. Int. J. Remote Sensing, Vol 19 (1988) 543-548

3. Stringer, W.J., Dean, K.G., Guritz, R.M., Garbeil, H.M., Groves, J.E., Ahlnaes, K.: Detection of Petroleum Spilled from the MV Exxon Valdez. Int. J. Remote Sensing, Vol. 13(5) (1992) 799-824.
4. Solberg, A. H. S., Storvik, G., Solberg, R., Volden,E.: Automatic Detection of oil Spills in ERS SAR Images. IEEE Trans. on Geoscience and Remote Sensing, Vol. 37(4) (1999) 1916-1924.
5. Liu A. K., Peng C. Y., Chang, S. Y.-S.: Wavelet Analysis of Satellite Image for Coastal Watch. IEEE Int. Journ. of Oceanic Engineering, Vol. 22(1) (1997)
6. Del Frate F. S., Petrocchi, A., Lichtenegger, J., Calabresi, G.: Neural Networks for Oil Spill Detection Using ERS-SAR Data. IEEE Trans. on Geoscience and Remote Sensing Vol. 38(5) (2000) 2282-2287
7. Malladi, R., Sethian, J.A.: Image Processing via Level Set Curvature Flow. Proc. Natl. Acad. of Sci., Vol. 92(15) (1995) 7046-7050
8. Mallat S. G.: A Wavelet Tour of Signal Processing. 2nd. Edn. Academic Press, Cambrige (1998)
9. Starck , J. L., Murtagh F. and Bijaoui, A.: Image Processing and Data Analysis: The Muliscale Approach. Cambridge University Press, Cambridge, UK.(1998).
10. Medeiros, F. N. S., Mascarenhas, N. D. A., Marques, R. C. P., Laprano C. M.: Edge Preserving Wavelet Speckle Filtering. Southwest Symposium on Image Analysis and Interpretation. Santa Fe, New Mexico (2002) 281□285
11. Gonzalez, R.C. and Woods R.E: Processamento de Imagens Digital. Edgard Bl□her Ltda, S□o Paulo, Brazil (2000)
12. Sita, G., Ramakrishnan, A.G.: Wavelet Domain Nonlinear Filtering for Evoked Potential Signal Enhancement. Computers and Biomedical Research Vol. 33(6) (2000) 431-446
13. Marques , R. C. P., Laprano, C. M., Medeiros, F. N. S.: Multiscale Denoising Algorithm Based on the □ Trous Algorithm. Proceedings of SIBGRAPI 2002 - XV Brazilian Symposim on Computer Graphics and Image Processing. Fortaleza, Cear□, Brazil (2002) 400
14. Lombardo P., Oliver C. J.: Optimum Detection and Segmentation of Oil-Slicks with Polarimetric SAR Data. IEEE International Radar Conference (2000) 122-127
15. Glasbey, C.A., Horgan, G.W.: Image analysis for the biologic sciences. John Wiley & Sons (1995) USA

Appendix : A

Formulas:

- Complexity= $\frac{\mathrm{P}}{2\sqrt{\pi}.\mathrm{A}}$
- Area to perimeter ratio $=\frac{\mathrm{A}}{\mathrm{P}}$
- Spreading is derived from the principal component analysis of the vectors whose components are the coordinates of the pixels belonging to the oil spill candidate. λ_1 and λ_2 are eigenvalues.

  $$\mathrm{Spr} = 100.\frac{\lambda_2}{\lambda_1 + \lambda_2}$$

Legend:

- A = area in pixels.
- P = perimeter in pixels.
- C.M. = center of mass.
- D_{max} = maximum distance, in pixels, from the boundary points to the center of mass.
- D_{min} = minimum distance, in pixels, from the boundary points to the center of mass.
- Spr. = spreading.
- Comp. = complexity.
- A/P = area to perimeter ratio (pixels).
- G_{max} = maximum value of the border gradient (dB).
- G_{me} = mean value of the border gradient (dB).
- Osd = object standard deviation (dB).
- Bsd = background standard deviation (dB).
- Gsd = standard deviation of the border gradient values (dB).
- C_{max} = maximum contrast, that is the difference between the background intensity mean value and the lowest value of the object.
- C_{me} = mean contrast, that is the difference between the background intensity mean value and the object mean value.

A New Edge-Grouping Algorithm for Multiple Complex Objects Localization

Yuichi Motai

Intelligent Media Laboratory
Department of Electrical and Computer Engineering
College of Engineering and Mathematics
University of Vermont
33 Colchester Avenue, Burlington, VT 05405, U.S.A.
ymotai@emba.uvm.edu
http://www.uvm.edu/~ymotai

Abstract. We present a new algorithm that provides an efficient localization method of elliptic industrial objects. Our proposed feature extraction inherits edge grouping approaches. But instead of utilizing edge linkage to restore incomplete contours, we introduce criteria of feature's parameters and optimize the criteria using an extended Kalman filter. Through a new parameter estimation under a proper ellipse representation, our system successfully generates ellipse hypotheses by grouping the fragmental edges in the scene. An important advantage of using our Kalman filter approach is that a desired feature can be robustly extracted regardless of ill-condition of partial edges and outlier noises. The experiment results demonstrate a robust localization performance.

1 Introduction

Robust localization of industrial objects for assembly tasks is an issue in the robot vision community. The difficulty of feature extraction for complex objects is never ameliorated under real assembly conditions. In order to achieve reliable automatic tasks, it is essential to develop robust feature extraction and its correspondence. Typical stereo views of a workspace are shown in Fig. 1, where automobile parts, called Alternator Cover, are our target objects to be manipulated by robot hand. In this paper, we propose a new algorithm that is capable of localizing these complex industrial parts by a single camera mounted on a robot hand that uses stereo views.

Numerical model-based vision systems have been developed to estimate object pose for robotic manipulation. In such algorithms [3],[13],[17], the correspondence search between model features and scene features needs to be solved first, and the precise estimation of 3D object pose needs to be then accomplished second. In our previous approach [9], for example, the features generated by region growing are used for matching first between the object scene and the correspondent model. However lack of perfect feature extraction occasionally fails the localization, due to lighting conditions, shading, occlusion, and sensory noise.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1194-1203, 2004.

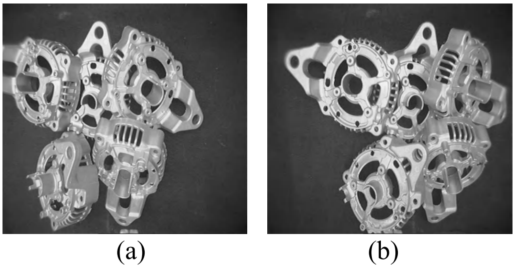


Fig. 1. Target objects of stereo views (a) left and (b) right.

A number of studies on this problem have shown that edge grouping methods [6] [7] are promising to extract salient features. Such algorithms aim to utilize global salient structures of object contours, inspired perceptual organization. Based on a set of edges, optimal curve fitting using Kalman estimation is an important extension [4], [15].

Our feature extraction method inherits these edge grouping approaches. In order to cope with partial occluded poses of the cluttered objects as shown Fig. 1, grouping is essential to estimate salient/high-level features from local/low-level edges. Regardless of the complexity of the object, edge grouping approaches are possible [7], and several partial edges are only cues to extract the salient features in severe conditions [4]. However when a target object is too complicated, it is difficult to extract features by other extraction methods such as region growing [9]. In the automobile assembly tasks, our target shape, ellipse is a large class of curved objects in industry. Currently our focus will be limited to 3D planar elliptical features, as exemplified by the curved silhouettes on the industrial object shown in Fig. 1. The first contribution in this paper is to extract salient ellipse feature to represent this complex object class through a new edge-grouping.

To detect elliptic geometrical features, the least mean square method is frequently used to fit the ellipse boundary [1], [3], although this method is very weak for outlier data. Main alternative methods are to utilize Hough transformation [5], [14] and moments method [16], although these methods handle to extract ellipses under some limited scattered images. Our approach handles noises and outliers by the extended Kalman filter to optimize specific ellipse parameters directly. Thus our second contribution is to derive a proper ellipse representation for a Kalman estimation under the new grouping algorithm to generate ellipse hypotheses.

In the following sections, we will first present the overall strategy for our object localization system. Then our main focus in this paper, feature extraction method by grouping edge will be presented in details. Subsequently feature matching will be described. Finally, experimental results with the system's evaluation will be shown.

2 Overall Strategy of System

In our feature-based bin-picking system, first of all the object model registration is required by a human-in-the-loop with a graphical editor through image-to-image and pose-to-pose correspondences [10]. After the 3D vision model is acquired in this manner, the same objects are randomly stacked. The goal of on-line localization system described here is robust 3D pose calculation of each object. Once the feature matching between model and scene is achieved, then the 3D translation and rotation from the model coordinate to the scene coordinate is computed using quaternion approach [2]. The robot gripper can pick up the object or perform a peg-in-hole alignment through its 3D localization.

In order to complete reliable matching results, the program first extracts smooth curves for fitting large ellipse, as seed features, based on the object model. For the object shown in Fig. 1, at least half numbers of holes among extracted regions should be matched in left and right images to look for an optimal solution based on the geometric constraint equation. The system generates the hypotheses of the large ellipse candidates, and then selects other holes which will verify the estimated pose. The verification can be done by looking for supporting features of the model, for example of this object, small holes.

Among many modules in our system, one of the main focuses in this paper is feature extraction module by a robust edge grouping technique. The feature extraction essentially determines robustness and accuracy of the object localization [9]. In our formalism, a *primitive* is defined as a salient entity of features by which matching between models and scenes can be directly established. For a representation of an industrial object called Alternator Cover, ellipses are such primitives. In the scene, however, such salient primitives may not be extracted perfectly. For example, the ellipses may be broken into smooth convex curves in the scene as shown in Fig. 2. Therefore we need to prepare immediate minimum entities extracted from images, and call such low-level/local image features for *fragments*.

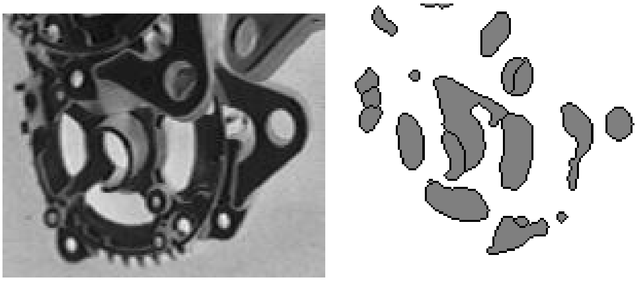


Fig. 2. False case of ellipses with broken smooth convex curves (a) a captured image (b) incomplete primitives.

Our edge grouping strategy is first to gather the fragments based on the attributes, such as size, convexity, and gray level similarity. The system then checks the elliptic curves by the number of fragments participating in forming the ellipses. For each group of fragments, the system estimates the parameters of hypothesized ellipses using iteration of Kalman filtering. In order to formalize the criteria of parameters, in the next section we will describe the representation of ellipse.

3 Representation of Ellipses

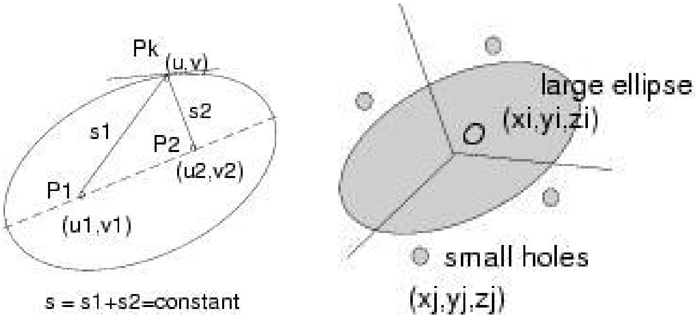
One of our contributions in this paper is a parametric representation and estimation of ellipses in the images which will be suitable for edge-based feature extraction. We reconsider this shortcoming of previous approaches [4], [15], and propose a proper parametric form for Kalman filter estimation. Although previous researchers represent ellipses by parametric equations of either

$$au^2 + buv + cv^2 + du + ev + f = 0 \tag{1}$$

or

$$\frac{(u\cos\theta - v\sin\theta - u_0)^2}{a^2} + \frac{(u\sin\theta + v\cos\theta - v_0)^2}{b^2} = 1\,. \tag{2}$$

where (u,v) be any arbitrary point on an ellipse in the image, a and b represent the lengths of the longer and shorter axes of the ellipse, θ the orientation of the ellipse, and (u_0, v_0) the center of the ellipse.



(a) (b)

Fig. 3. (a) Ellipse model (b) Seed feature (3D ellipse) and supporting features (small holes).

The first representation (1) includes arbitrary quadratic forms other than ellipses, and therefore it is difficult to deal with geometric constraints over an ellipse by a set of parameters (a,b,c,d,e,f). Because the parameters are no explicitly bounded, - wide range of parameters - there is no guarantee to generate elliptic curve.

The second representation Eq. (2) is more intuitive, but we have observed the instability of estimating the orientation parameter θ when a and b are getting close, since the orientation becomes arbitrary as a regular circle. The parameters (a,b,u_0,v_0,θ) cause the serious problem of an instable extraction.

We have exploited a different approach to represent ellipses which will be useful to estimate all proper parameters of elliptic features extracted from the image. As shown in Fig. 3, we represent the ellipse by two focal points P_1 and P_2 and the sum s of the distances (s_1, s_2) from the two focal points to any boundary points P_k. Let (u_1, v_1) and (u_2, v_2) be image coordinates of two focal points P_1 and P_2, and (u, v) be the image coordinate of arbitrary boundary point P_k. Then

$$f \equiv \sqrt{(u-u_1)^2+(v-v_1)^2}+\sqrt{(u-u_2)^2+(v-v_2)^2}-s=0 \tag{3}$$

Our contribution using Kalman filter approach includes this proper ellipse representation to derive criterion function. The parameterization in Eq. (3) is very important when we apply Kalman filtering. In this specific parametric representation, if two focal points get close, the two focal points become simply coincident; therefore no instability for the parameter estimation can be observed. So our problem is how to estimate five ellipse parameters $\boldsymbol{p}=(u_1, v_1, u_2, v_2, s)$ from primitives extracted from the images.

4 Feature Extraction: Edge Grouping

Our feature extraction starts from 2D image processing, in which the local edges are extracted as fragments, described in subsection 4.1. Subsequently the group of fragments is generated as hypothesis of ellipse, described in 4.2.

4.1 Extraction of Fragments

After Canny edge detector is applied to an image, the edges are finely sampled to local segments--called *fragments*. These sampled edges are defined by tracking edges. Fragments along curves in the 2D image scene are automatically generated by the system in the following manners:

1. Thin the edge pixels so that the edge tracking can be performed.
2. Extract endpoints and junction points in the thinned edge map.
3. Track the edge map and extract high curvature points along edge curves.
4. Divide the long smooth curves into at least two components to avoid the accidental coincidence of merged curves.
5. Register curve segments as fragments.

Note that the selection of high curvature points is done by smoothing the curve along its original form. If the deviation of the smoothed curve from the original curve is higher than some threshold and is maximal, then the system registers these points as high curvature points. As a base of low-level feature, the system decomposes the curve into smaller pieces if the curve is long enough and occupies a large angle for the ellipse formation. The decomposing is very useful for avoiding accidental coincidence, by chance, two different curves are merged due to the viewpoint ill-conditions.

4.2 Extraction of Ellipse Hypotheses

Fig. 4 illustrates the procedure to extract ellipse candidates in the scene image based on grouping fragments. The system checks the elliptic curves by the number of fragments participating in forming the ellipses. As we have discussed in the previous section, a single curve extracted from the image may not necessarily correspond to a perfect ellipse. An ellipse may be broken into several fragments. Therefore, we deal with the grouping of the fragments which potentially constitute an ellipse. The grouping of fragments is decided on the following constraints:

1. *size:* The size of the ellipse in the image is limited. For each group of fragments, the combined curves must be smaller in size than some threshold based on the object model.

2. *convexity:* Any pair of fragments must not violate the convexity when these fragments are combined.

3. *gray level similarity:* Any pair of fragments must possess gray level similarity. Either internal or external region has the similar gray level. Note that this is a typical case for an industrial object when the object is composed of parts of homogeneous color. If the object region is homogeneous along the elliptic curve, then two fragments *i* and *j* must satisfy the gray level similarity constraint:

$$\frac{|\mu_i^{Internal}-\mu_j^{Internal}|}{\sigma_i^{Internal}+\sigma_j^{Internal}}<\varepsilon \quad \text{or} \quad \frac{|\mu_i^{External}-\mu_j^{External}|}{\sigma_i^{External}+\sigma_j^{External}}<\varepsilon \tag{4}$$

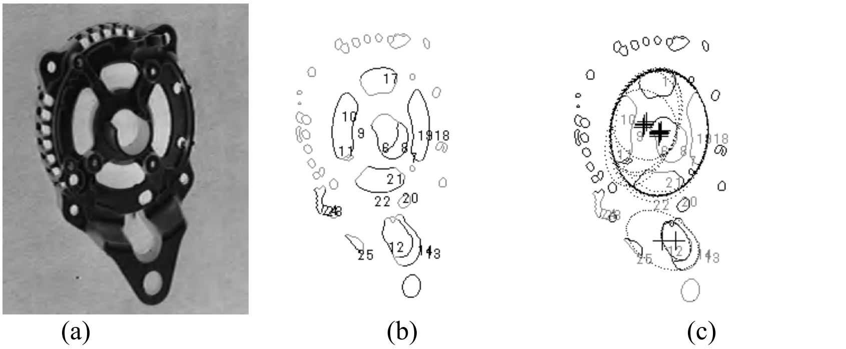


Fig. 4. Edge grouping sequential process (a) Object scene, (b) Curve fragments extraction along edges, (c) Group of fragments extracted by aggregation through elliptical parameter estimation.

In our current implementation, the system generates an ellipse hypothesis based on how many fragments are chosen for grouping, from a single fragment to four fragments. The system first estimates initial parameter of each ellipse and then updates for verifying that ellipse. For each group of fragments for an ellipse candidate, the system verifies whether or not these fragments certainly constitute an ellipse in terms of the parameters. More specific procedure is as follows:

- **Generation of Initial Parameter Estimation**

Given a set of points along group of the fragment curves, $\boldsymbol{p} = (u_1, v_1, u_2, v_2, s)$ is to be estimated. First of all, the system generates an initial estimate of $\boldsymbol{p}$, and then applies the Kalman filter to update the parameter $\boldsymbol{p}$ [8]. In order to compute the initial estimate of $\boldsymbol{p}$, we first compute the centroids and the moment of inertia for the image points participating in the fragment set. The initial estimates of (u_1, v_1) and (u_2, v_2) are computed as the above centroids. The initial estimate of s is computed by the sum of the lengths of two axes spanned by the moment of inertia. We also associate the covariance matrices for these parameters. The covariance matrices are assigned on the basis of the experiments.

- **Verification of Ellipse Formation**

After the system obtains the initial estimate of the ellipse parameter $\boldsymbol{p}$, the system selects representative points from the fragments. This is done by equally selecting points along the boundary curves. In our current implementation, the system selects at least 16 points for each fragment. By applying the Kalman filter to the constraint equation of Eq. (3) for every selected boundary point (u, v), the system updates the ellipse parameter $\boldsymbol{p}$.

5 Feature Matching

We utilize an object model, consisting of ellipse (seed feature) and small holes (supporting feature) illustrated in Fig. 3 (b). In the first step, the 2D results of edge grouping extraction (described in Section 4.2) are reconstructed in 3D. Hypothesis generation of objects based on seed features (Large Ellipses) are used for matching as following procedures:

i. Given groups of fragments, generate hypothesis that optimally fits to the each hole by computing elliptic parameters.

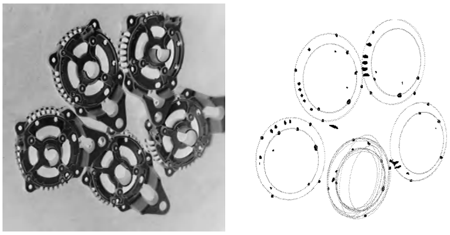
ii. For each hypothesized ellipse in the left, look for an ellipse in the right image which will correspond to the left one by considering the epipolar constraint.

iii. By epipolar constraint, estimate the seed feature position in the 3D space. Check whether or not this feature will support the estimated pose of the object.

iv. Apply attribute constraints of the correspondent model, such as area, circularity, shape complexity, perimeter, and average gray level with deviation, many hypothesized regions are pruned out.

As you expect, there exist several mismatches of generated hypothesis, which should be removed by verification using supporting features. In the second step, hypothesis verification of objects is used for supporting features (Small Holes).

Two fine-line ellipses in Fig. 5 (b) represent where the small holes should be given the estimated poses. These supporting features are extracted by a different extraction method, such as region-based method called split and merge segmentation process [9], because this method is just useful to extract small holes as you see Fig. 5(b). The second steps are described more details as follows:



(a) (b)

Fig. 5. (a) 2D captured view (b) Small holes extracted along hypothesized regions.

i. Based on the 3D object pose hypothesis, generate constraint regions of 3D supporting features where the supporting features should be in the left and right images, by projecting the 3D supporting features onto the 2D images with a given pose associated with the hypothesis.
ii. Select 2D features associated with 3D supporting features in the left and right images within the constraint regions.
iii. Estimate the 3D positions of 3D supporting features based on the stereo correspondences of 2D features obtained in previous step **ii**.
iv. Verify the hypothesis by considering the geometric constraints of 3D supporting features: These 3D supporting features should be compatible with the hypothesized 3D object pose given in step **i**, as well as the 3D supporting features satisfy the geometric constraints, e.g., distance between 3D supporting features, orientation between 3D supporting features. Also the number of supporting features should exceed a user specified threshold.
v. Find an optimal solution, if multiple solutions exist within a certain portion of the workspace. (This may happen due to the edge grouping). For each solution, we approximate the object space occupancy. If multiple solutions share the space occupancy, then select the optimal solution from such shared solution sets. The optimality is based on the geometric constraints of the 3D seed feature and 3D supporting features, which is associated with the fitting error.

6 Experimental Results

We mounted a monocular camera on the robotic manipulation gripper to generate a 3D object model by capturing multiple images of the object from different viewpoints. Our wrist-mounted robotic vision systems consisted of a Sony *DC-47* monocular *1/3 inch* CCD camera with Pulnix Lens of focal-length *16 mm*, a Kawasaki *JS-10* and a PC. We used only two views to localize the object for automatic localization. Curve shaped *4 - 6* pieces objects, automobile industrial Alternator Cover, were used for automatic localization experiments, *35* times, total *144* pieces. The typical 3D translation errors were less than *3 mm* and rotation errors were less than *5 degree*. The outline diameter was *65 mm* for Alternator Cover of symmetric circle outline. These localization results were verified through the robotic manipulation. For example, the error of *4 mm* and *8 degree* was within the tolerance range for our robot manipulation of that Alternator Cover. Since we already calibrated all the robotic coordinate transformations [11], the robot hand can manipulate the localized object shown in Fig. 6 (e) (f). Fig. 6 (a) (b) shows typical results of the edge grouping of left and right 2D image, and (c) (d) shows the 3D model projected onto the 2D object scene, -- bold ellipses represent the estimated poses from all candidates, which seems to graphically be lined up well.

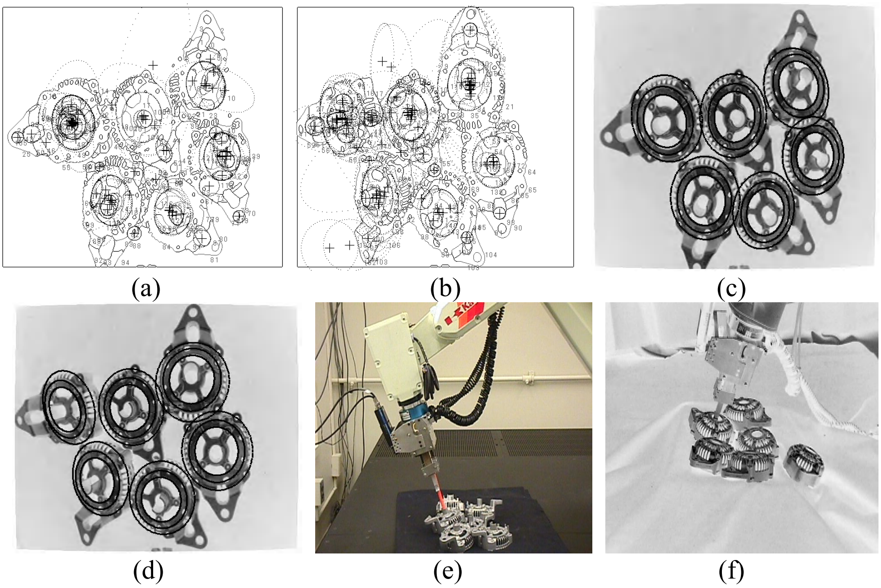


Fig. 6. (a) A typical experimental result of Alternator Cover localization (a) left view of edge grouping, (b) right view of edge grouping, (c) left view of 2D superimposition of model onto scene, (d) right view of 2D superimposition of model onto scene, (e) robot manipulation demonstration from far view, (f) robot manipulation demonstration from close view.

We also evaluated the success rates of the system into the following categories:

- successful localization case: localization completed with a success robotic manipulation (within tolerance range) [82.6%]
- inaccurate localization case: localization completed with false manipulation (outside tolerance range) [1.4%]
- incomplete localization case: no localization output [16.0%]

7 Conclusions

We developed a new vision-based localization system for 3D elliptic industrial objects. Our salient feature extraction from complex objects extended traditional edge grouping approaches. More specifically, our contributions of this system were (1) to establish a competent edge-grouping method to generate ellipse hypotheses in a complex object and

(2) to derive an efficient ellipse representation for Kalman estimation. Using each group of edge fragments, the system estimated the parameters of ellipse hypotheses using an extended Kalman filtering. The advantage of Kalman estimation was that a desired feature could be robustly extracted regardless of ill-condition of partial occlusions and outlier noises. For optimizing the criterion, we introduced a proper parametric representation of an ellipse feature to achieve a stable result. The evaluation experiments verified that our feature extraction and its matching method were robust for an object manipulation.

References

1. Chatterjee, C. and Chong: E.K.P., Efficient algorithms for finding the centers of conics and quadrics in noisy data, Pattern Recognition, vol. 30, (1997) 673-684
2. Faugeras, O.D.: Three-Dimensional Computer Vision, MIT Press, (1993)
3. Fitzgibbon, A, Pilu, M. and Fisher, R. B.: Direct Least Square Fitting of Ellipse, IEEE Trans. of Pattern Analysis and Machine Intelligence, Vol. 21, (1992) 476-480
4. Guichard, F. and Tarel, J: Curve finder combining perceptual grouping and a Kalman like fitting, Proc. of IEEE International Conf. on Computer Vision, Vol. 2, (1997) 1003-1009
5. Hough, P.V.C.: Method and means for recognizing complex patterns, US Patent, No. 3 069 654, (1962)
6. Huttenlocher, D. P. and Wayner, P.C.: Finding convex edge groupings in an image, International Journal of Computer Vision, Vol. 8, (1992) 7-27
7. Jacobs, D.W.: Robust and efficient detection of salient convex groups, IEEE Trans. on Pattern Analysis and Machine Intelligence, 4:30 PM 10/31/2003Vol. 18, (1996) 23-37
8. Kosaka, A and Kak, A.C.: Fast Vision-guided Mobile Robot Navigation using Model-based Reasoning and Prediction of Uncertainties, Computer Vision, Graphics, and Image Processing - Image Understanding, Vol. 56, (1991) 271-329
9. Kosaka, A. and Kak, A.C.: Stereo Vision for Industrial Applications, Handbook of Industrial Robotics, edit. S. Y. Nof, John Wiley & Sons, Inc., (1999) 269-294
10. Motai, Y and Kak, A.C.: An Interactive Framework for Acquiring Vision Models of 3D Objects, IEEE Trans. of Systems, Man, and Cybernetics - Part B: Cybernetics, in press, (2003)
11. Motai, Y and Kosaka, A: SmartView: Hand-Eye Robotic Calibration for Active Viewpoint Generation and Object Grasping, Proc. of IEEE Int. Conf. of Robotics and Automation, (2001) 2183-2190
12. Motai, Y and Kosaka, Y.: Concatenate Feature Extraction for Robust 3D Elliptic Object Localization, Accepted in 19th ACM Symposium on Applied Computing, AI track, (2004)
13. Niwakawa, M., Onda, T. and Fujiwara, N., Vision-based handling system using model-based vision and stereo ranging, 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, Vol. 2, (1998) 199-204
14. Olson, C.F.: Improving the generalized Hough transform through imperfect grouping, Image and Vision Computing, Vol. 16, (1998) 627-634
15. Porrill, J.: Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter, Image and Vision Computing, Vol. 8, (1990) 37-41
16. Voss, K. and Suesse, H.: Invariant Fitting of Planar Objects by Primitives, IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol. 19, (1997) 80-84
17. Yoshimi, B.H. and Allen, P.K.: Active uncalibrated visual servoing, Proceeding of IEEE International Conference on Robotics and Automation, Vol. 1, (1994) 156-161

Processing and Analysis of Ground Penetrating Radar Landmine Detection

Jing Zhang[*] and Baikunth Nath

Department of Computer Science & Software Engineering,
The University of Melbourne, Melbourne, Vic 3010, Australia
{jzhang, bnath@cs.mu.oz.au}

Abstract. There are mainly two types of landmine, anti-tank mine (ATM) and anti-personnel mine (APM), which kill or maim people around the world. Much research effort has been devoted in trying to detect and remove APM. APM is smaller than ATM and in addition, most of them are made of plastic making them more difficult to be detected. Some of the common mine detection techniques include using sensors to get raw data, signal processing and image processing. Plastic APM has dielectric properties, this makes it similar to that of soil and the reflection from the mine is usually weak and masked by background. It is therefore very difficult to detect this type of landmine with conventional Ground Penetrating Radar (GPR), though it is a good method to detect metallic landmine. Many new techniques have appeared in recent years in an effort to solve this so called "inverse scattering problem". In this paper, we review and discuss the GPR data acquisition, signal processing, image processing methods.

1 Introduction

Landmines kill and maim approximately 26,000 people annually. Direct casualties are not the only problem. In some place, whole areas of arable land cannot be farmed due to the threat of landmines [1].

Current demining techniques are heavily reliant on metal detectors and prodders. In many circumstances, the prodder is the first and the last resort. The advent of non disturbance fused mines makes prodding a dangerous operation. Mechanical devices such as ploughs, rollers, and flails are usually followed by manual demining to obtain the desired level of clearance. These machines are expensive for developing countries. Dogs are good when they work but can only operate for limited periods and must be acclimatized. In order to assist deminers, a range of advanced sensor technologies are being investigated [2], including

- metal detectors (MD) □ capable of finding even low-metal content mines in mineralized soils;

[*] On leave from the Computer Science and Technology Engineering Institute, Harbin Engineering University, China.

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1204-1209, 2004.

- nuclear magnetic resonance, nuclear quad rapole resonance (NQR), fast neutron activation and thermal neutron activation □ detect the presence of the explosive material in landmines;
- thermal imaging (TI) and electro-optical sensors (EO) □ detect evidence of a buried object, such as disturbed ground or the thermal effect of having a mine just below the surface;
- biological sensors such as dogs, pigs, bees, and birds;
- chemical sensors, such as thermal fluorescence and chromatographic techniques □ detect airborne and water borne presence of explosive vapors.

In this paper, we will concentrate on Ground Penetrating Radar (GPR). This ultra wide band radar provides centimeter resolution to locate even small targets. GPR operates by detecting the dielectric contrasts in the soils, which allows it to locate even nonmetallic mines. Unfortunately, this technology can suffer false alarm rates as high as that of metal detectors. We shall discuss some advanced GPR methods.

2 GPR Data Acquisition

There are two distinct types of GPR, time-domain and frequency domain [3]. Time domain or impulse GPR transmits discrete pulses of nanosecond duration and digitizes the returns at GHz sample rates. Frequency domain GPR systems transmit single frequencies either uniquely, as a series of frequency steps, or as a chirp. The amplitude and phase of the return signal is measured. The resulting data is converted to the time domain. But the procedure of getting the raw GPR data of both types is similar. The description of the method follows.

2.1 GPR Data

GPR consists of an active sensor, which emits electromagnetic (EM) waves through a wideband antenna and collects signals reflected from its surroundings. The principle of GPR is almost the same as in a seismic wave measurement system except for the carrier signal. The commonly used frequency band of the GPR, EM wave is between 100 MHz and 100 GHz [4].

This band is wide enough to carry the necessary information. Reflection occurs when the emitted signal encounters a surface between two electrically different materials. The direction and intensity of the reflection depend on the roughness of the surface and electrical properties of the medium material.

A rough surface reflects the incident wave in a diffused manner, while a smooth surface tends to reflect the wave in one direction, where the angle between the surface (normal) and the reflected wave is the same to the angle between the surface (normal) and the incident wave. The electrical property of the medium determines the amount of refraction and absorption of the EM waves and subsequently affects the direction and intensity of the reflection.

The penetration depth of the wave into soil usually depends on two factors, the humidity in the soil and the wavelength of the EM wave. The content of water in the soil significantly reduces the depth of penetration of a wave with relatively shorter wavelength. Based on the reflection and penetration properties, GPR works best with low-frequency EM waves in dry sand. Low-frequency signals, however, tend to make low-resolution maps of data, which decreases the accuracy of mine detection. Since the EM waves cannot penetrate water, GPR cannot detect underwater mines, which are common in many countries.

GPR provides information on both the existence and location of mines. The presence of an object is detected by checking for interruption through the round trip path of the signal. The distance between the sensor and an object is ensured by using the time delay Δt, between the emitting and receiving moments of the signal as

$$R = (v/2)\Delta t \tag{1}$$

Where v represents the velocity of the EM wave in the medium, and R is the distance of the object from the sensor. Since many parameters of the EM waves, including the velocity, vary according to the content of soil, soil parameters should be estimated prior to taking the measurement [5].

2.2 A, B, and C-Scan

GPR data can be represented in three different forms, A, B, and C - Scans, according to the scanning dimension. It can be defined with the 3D coordinate system, where the x y-plane represents the ground surface and the z-axis represents the direction into the ground.

The A-scan signal is obtained by a stationary measurement after placing an antenna above a specific position. The collected signal is presented in the form of a group of signal strength versus time delay. A-scanned signal measured at the position (x', y') is a 1D signal. B-scan signal is obtained as the horizontal collection from the ensemble of A-scans. The collected signal is presented as intensity on the plane of scanned width versus time delay. Therefore, the B-scanned signal can be considered as a 2D signal. C-scan signal is obtained from the ensemble of B-scans, measured by repeated line scans along the plane. The collected C-scanned signal forms a 3D signal. In the 3D coordinate system (Figure 1), the x and y axes respectively represent the horizontal and the vertical positions of the target, and the z-axis represents the depth of the target.

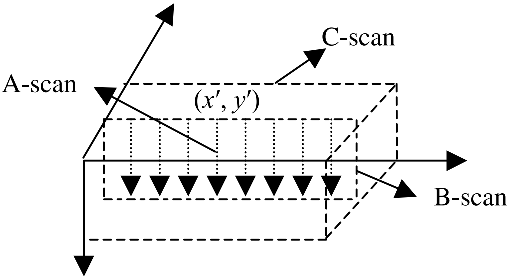


Fig. 1. The 3D coordinate system defined on a section of ground

Since visualization of a three-dimensional data is not easy, a C-scan is usually represented by a collection of horizontal slices for a specific data point, that is *x y*-planes at each specific position on the *z*-axis. Each slice corresponds to a certain depth level, which is equivalent to the vertical axis of the B-scan.

3 Preprocessing

In general for mine detecting processes, 2D information for mine location is the most important. Independent of the sensor used, the sensor output can be represented in the form of 2D data, which can be considered an image. A slice in the ensemble of C-scanned GPR data can be considered as a 2D image, where local contrast in pixel intensity provides a clue for potential existence and location of mines. 2D data from the sensors are highly subject to degradation due to various factors, such as (i) noise due to unpredictable combination of soil contents, (ii) low-resolution due to the limited performance of a sensor, and (iii) low-contrast due to the limited dynamic range of the sensor output. For these reasons, the data must be enhanced by using various signal and image processing techniques.

Usually, there are subsequent steps to detect landmine with signal processing. In the first step, obtain a rough picture by some methods, such as using a simple filtered time-domain back propagation algorithm. In the second step, apply the model to recover further details about the buried target. In the third step, the typical using is the inversion method to retrieve more details of the inhomogeneous character of the buried objects. The homogeneous character problem is called the inverse scattering.

Inverse scattering includes two problems aspects. One is the problem in spatial-domain, the other one is in the time domain. There are some common methods to solve the inverse scattering problems. For example, in spatial-domain, there are the finite-element method and the Polak□Ribi□re nonlinear conjugate gradient optimization algorithm. In the time domain, there are the filtered time-domain backpropagation, the Finite-Difference Time-Domain method (FDTD method), time-domain electromagnetism, the forward-backward time-stepping (FBTS) algorithm and gradient-based optimization method.

However, the above methods can not cope with these Inverse scattering problems, such as reconstructing the electromagnetic properties of unknown scatterers, reconstruct its internal properties, transient electromagnetic field problems and the characterization of a buried object.

Rekanos [6] has given a method to solve the problem of reconstructing the electromagnetic properties of unknown scatterers and is treated by means of a spatial-domain technique. He combines the finite-element method and the Polak□Ribi□re nonlinear conjugate gradient optimization algorithm and solves a forward scattering problem by the finite-element method. What□s more, he use a standard error term and a regularization term to implement the inversion by minimizing a cost function. The standard error term is related to the scattered near-field measurements, which are obtained by illuminating the scatterer with plane waves from various directions of incidence [7]. The regularization term is introduced in order to cope with the ill-posedness of the inversion.

The filtered time-domain backpropagation can give a good indication of the buried object's boundary. However, fails to reconstruct its internal properties (constitution). Budko [8, 9] exploits a different way of restricting the class of solutions by changing the scale of the problem. First, an image of the scattering domain is computed using a simple filtered time-domain backpropagation algorithm. The result of this stage is a rough picture of spatial variations of constitutive parameters in the subsurface. Usually, this indicates the sharp changes in the constitution, i.e. boundaries of objects. Next, the outlined objects are presumed to be homogenous, and the effective scattering model is constructed using interactive graphical software [10].

In the electromagnetics community, the Finite-Difference Time-Domain method (FDTD method) [11] is the main tool to solve transient electromagnetic field problems. In this method, the electromagnetic field quantities are approximated in space on a staggered grid (Yee-mesh) and the time discretization is carried out in a leap-frog manner. The Courant-Friedrichs-Lewy stability condition for FDTD is well known, but this condition is necessary only [12]. They have derived a condition that is both necessary and sufficient. Up till now the stability analysis is restricted to lossless media. Presently, they are trying to include lossy media as well [13]. For configurations with high conductivity values, this method is much more efficient than the well-known Finite-Difference Time-Domain method. A necessary and sufficient condition for stability of the latter method has also been derived. In addition, they have used the reduced-order modeling technique [14] to simulate experiments with a Ground Penetrating Radar (GPR) and to characterize the shallow subsurface of the Earth.

4 Conclusions

We have reviewed and discussed some of the more recent signal and image processing techniques that have been applied to the mine detection area. In addition, some GPR data acquiring technologies are also stated.

This paper introduced how to get the raw GPR data and described the three types of the raw data. 2D data from the sensors often is low-resolution and low-contrast and with noise. This often leads to the homogeneous character problem (the inverse scattering problem). So in part of the processing, we introduce the Rekanos method to reconstruct the electromagnetic properties of unknown scatterers; for the object's boundary problem, we have introduced Budko method. Finally, for the transient electromagnetic field problems, we introduced the FDTD method.

Many research groups have developed new detection devices with multiple sensors, and also the corresponding technology, called sensor fusion, to combine outputs from multiple sensors. This survey will serve as a signal and image processing background to better aid in understanding of existing technologies and in developing new technologies for mine detection.

References

1. Zoubir A M, Chant I A and Brown C I: Signal Processing Techniques for Landmine Detection Using Impulse Ground Penetrating Radar. IEEE Sensors Journal, Vol. 2, pp 41-51, 2002.
2. Paik J, Lee C P and Abidi M A: Image Processing-Based Mine Detection Techniques: A Review. Subsurface Sensing Technologies and Applications, Vol. 3, pp 153-202, 2002.
3. Pizurica A, Philips W, Lemahieu I and Acheroy M: The application of a Nonlinear Multiscale Method to GPR Image Processing. Proceedings of the IASTED International Conference on Signal and Image Processing (SIP'98), Las Vegas, Nevada, USA, pp 332-335, 1998.
4. Jong W de, Lensen H A, and Janssen Y H: Sophisticated test facility to detect land mines. Detection and Remediation Technologies for Mines and Minelike Targets IV (SPIE Proceedings), Vol. 3710, pp.1409-1418, 1999.
5. Yarovoy A G , Lighart L P, Schukin A D and Kaploun I V: Full-polarimetric video impulse radar for landmine detection. Ninth International Conference on Ground Penetrating Radar (SPIE Proceedings), Vol. 4758, pp 246-250, 2002.
6. Rekanos I T, Efraimedou M S, Yioultsis T V, Antonopoulos C S and Tsiboukis T D: Microwave Imaging of Two-Dimensional Scatterers by Inverting Scattered Near-Field Measurements. Fifth International Symposium on Antennas, Propagation and EM Theory (ISAPE2000), Beijing, China, pp187-190, 2000.
7. Takenaka T, Jia H and Tanaka T: Microwave imaging of electrical property distributions by a forward-backward time-stepping method. Journal Electromagnetic Waves Applications, Vol. 14, pp 1611-1628, 2000.
8. Budko N V and van den Berg P M: From imaging to inversion: extending the capabilities of a standard GPR. International Symposium on Electromagnetic Theory (URSI Proceedings), Victoria, Canada, pp 456-458, 2001.
9. Budko N V and van den Berg P M: Characterization of a two-dimensional subsurface object with an effective scattering model. IEEE Trans. Geoscience and Remote Sensing, Vol. 37, pp 2585-2596, 1999.
10. Weldon T, Gryazin Y, Klibanov M: Novel Inverse Methods in Land Mine Imaging. ICASSP 2001, Salt Lake City, UT, USA, 4 pages, 2001.
11. Budko N V, Remis R F and van den Berg P M: Numerical Modeling of Ground Penetrating Radar. Part 2: Imaging and Effective Inversion. Proceedings International Conference on Electromagnetics in Advanced Applications (ICEAA'99), Torino, Italy, pp. 79-82, 1999.
12. Remis R F and van den Berg P M: Computing the electromagnetic field in a perturbed configuration using modified reduced-order models. Proceedings of the Sixteenth Conference on Annual Review of Progress in Applied Computational Electromagnetics, Monterey, CA, U.S.A., Vol. 1, pp 93-98, 2000.
13. Roth F, van Genderen P and Verhaegen M: Radar response approximations for buried plastic landmines. Ninth International Conference on Ground Penetrating Radar (SPIE Proceedings), Vol. 4758, pp.234-239, 2002.
14. Groenenboom J, and Yarovoy A G: Data processing for a Landmine detection dedicated GPR. Proceedings of the Eight International Conference on Ground Penetrating Radar (GPR2000), Gold Coast, Australia, May 2000.

Tuning of Power System Stabilizers via Genetic Algorithm for Stabilization of Power Systems

Farzan Rashidi[1] and Mehran Rashidi[2]

[1] Control Research Department, Engineering Research Institute, Tehran, Iran,
P.O.Box: 13445-754, Tehran
f.rashidi@ece.ut.ac.ir
[2] Hormozgan Regional Electric Co. Bandar-Abbas, Iran,
P.O.Box: 79167-95599, Bandar-Abbas
mrashidi@mehr.sharif.edu

Abstract. The paper considers simultaneous placement and tuning of power system stabilizers for stabilization of power systems over a wide range of operating conditions using genetic algorithm. The power system operating at various conditions is considered as a finite set of plants. The problem of setting parameters of power system stabilizers is converted as a simple optimization problem that is solved by a genetic algorithm and an eigenvalue-based objective function. A single machine –infinite bus system and a multi-machine system are considered to test the suggested technique. The optimum placement and tuning of parameters of PSSs are done simultaneously. A PSS tuned using this procedure is robust at different operating conditions and structure changes of the system.

1 Introduction

Much effort has been invested in recent years, in the development of power system stabilizers (PSSs) for improving the damping performance of power systems. The requirement for improved damping has arisen from a number of factors, including the development of high speed excitation systems, the use of long high-voltage transmission lines, and improvements in the cooling of turbo-alternators [9, 4]. The application of genetic algorithm (GA) has recently attracted the attention of researchers in the control area [2, 7, 8]. Genetic algorithms can provide powerful tools for optimization. In this work the structure of PSS is imposed and search is done on the parameters of the PSS by GA. The use of high-speed excitation systems has long been recognized as an effective method of increasing stability limits. Static excitation systems appear to offer the practical ultimate in high-speed performance thereby providing a gain in stability limits. Unfortunately, the high speed and gains that give them this capability also result in poor system damping under certain conditions of loading [6]. To offset this effect and to improve the system damping, stabilizing signals are introduced in the excitation systems through fixed parameters lead/lag PSSs [9]. The parameters of the PSS are normally fixed at certain values which are determined under a particular operating condition. It is important to recognize that

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1210-1219, 2004.

machine parameters change with loading, making the dynamic behavior of the machine quite different at different operating points [1]. So a set of PSS parameters that stabilizes the system under a certain operating condition may no longer yield good results when there is a change in the operating point. In daily operation of a power system, the operating condition changes as a result of load changes. The power system under various loading conditions can be considered as a finite number of plants. The parameters of the PSS that can stabilize this set of plants can be determined offline using a genetic algorithm and an objective function based on the system eigenvalue. Genetic algorithms are used as parameter search techniques, which utilize the genetic operators to find near optimal solutions. The advantage of the GA technique is that it is independent of the complexity of the performance index considered. The PSS designed in this manner will perform well under various loading conditions and stability of the system is guaranteed. However, the conventional PSS will only perform well at one operating point. The system to be studied is:

A. A single machine connected to an infinite bus through a transmission line.

B. A three machines system. Two kinds of PSS are considered. Derivative type power stabilizer and lead speed stabilizer with washout filter.

2 System Model

The system that described before is shown in fig.1. The synchronous machine is described by Heffron- Philips model. The relations in the block diagram when using derivative power stabilizer is shown in figure 2 apply to two-axis machine representation with a field circuit in the direct axis but without damper windings. The interaction between the speed and voltage control equations of the machine is expressed in terms of six constants K_1-K_6. These constants with the exception of K_3 which is only a function of the ratio of the impedance depend on the actual real and reactive power loading as well as the excitation system in the machine [5].

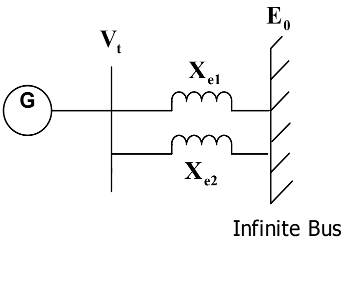


Fig. 1. Single machine connected to infinite bus

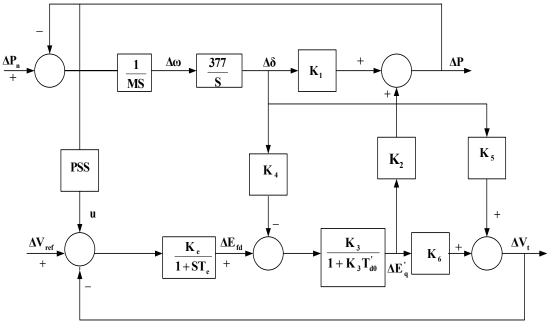


Fig. 2. System block diagram

The equations describing the steady-state operation of synchronous generator connected to an infinite bus through an external reactance can be linearized about any particular operating point as follows:

$$\Delta P_m - \Delta P = Md^2\Delta\delta / dt^2 \tag{1}$$

$$\Delta \mathrm{P} = \mathrm{K}_1\Delta\delta + \mathrm{K}_2\Delta \mathrm{E}^{'}q \tag{2}$$

$$\Delta E^{'}q = \frac{K_3\Delta E_{fd}}{(1+ST^{'}_{do}K_3)} - \frac{K_3K_4\Delta\delta}{(1+ST^{'}_{do}K_3)} \tag{3}$$

$$\Delta V_t = K_5\Delta\delta + K_6\Delta E^{'}q \tag{4}$$

The constants K_1-K_6 are given in section 5. The system parameters are as follow:

A: Machine Parameters (pu):

$$x_d = 1.6;\ x^{'}_d = 0.32;\ x_q = 1.55\ v_{t0} = 1.0;\ \omega_0 = 120\pi\ rad/\mathrm{sec};\ T^{'}_{do} = 6\,\mathrm{sec} \\ D = 0.0,\ M = 10.0 \tag{5}$$

B: Transmission line (pu)

$$r_e = 0.0,\ x_{e1} = x_{e2} = 0.4,\ x_e = x_{e1}\|x_{e2} = 0.2 \tag{6}$$

C: Exciter

$$K_e = 50 \qquad T_e = 0.05Sec \tag{7}$$

D: Loading (pu)

$$P = (0.1,0.2,...,1); Q = (-0.2,-0.1,...,1) \tag{8}$$

The stabilizing signal considered is:

A: proportional to electrical power and a derivative-type power stabilizer with the transfer function given by:

$$G_s(S) = K\frac{S}{(S+\frac{1}{T})^2} \tag{9}$$

Where *K* and *T* are the PSS parameters to be selected proportional to speed of rotor and a lead stabilizer and washout filter with the transfer function given by:

$$G_s(S) = K_c\frac{ST}{1+ST}\frac{(1+ST_1)}{(1+ST_2)} \tag{10}$$

Where K_c, T_1 and T_2 are the PSS parameters to be selected. The washout time constant *T* is considered 2 seconds.

3 Fitness Function

The problem of tuning the parameters of a single PSS for different operating points means that PSS must stabilize the family of *N* plants:

$$\dot{x}(t) = A_k x(t) + B_k u(t),\ k = 1,2,...,N \tag{11}$$

Where $x(t) \in R^n$ is the state vector and *u(t)* is the stabilizing signal. A necessary and sufficient condition for the set of plants in equation (10) to be simultaneously stabilizable with stabilizing signal is that eigenvalues of the closed-loop system lie in the left- hand side of the complex s-plane. This condition motivates the following approach for determining the parameters *K* and *T* of the PSS:
Selection of *K* and *T* to minimize the following objective function:

$$J = \max \mathrm{Re}(\lambda_{k,l}), k = 1,...,N, l = 1,...,N \tag{12}$$

Where $\lambda_{k,l}$ is the *l*th closed-loop eigenvalue of the *k*th plant, subject to the constraints that $K<a$ and $T<b$ for appropriate prespecified constant *a* and *b*. clearly if a solution is found such that $J<0$, then the resulting *K* and *T* stabilize the collecting of plants. The existence of a solution is verified numerically by minimizing *J*. The optimization problem is easily and accurately solved using genetic algorithms.

4 Genetic Algorithm

Genetic algorithms (GA) have been used to solve difficult problems with objective functions that do not possess well properties such as continuity, differentiability, etc., these algorithms maintain and manipulate a population of solutions and implement the principle of survival of the fittest in their search to produce better and better approximations to a solution. This provides an implicit as well as explicit parallelism that allows for the exploitation of several promising areas of the solution space at the same time. The implicit parallelism is due to the schema theory developed by Holland, while the explicit parallelism arises from the manipulation of a population of points. The power of the Genetic Algorithms (GA) comes from the mechanism of evolution, which allows searching through a huge number of possibilities for solutions.

5 Simulation Results

A…Unstabilized system (u=0)
Without any stabilizing signal, the system equation can be expressed in the following state variable form:

$$\dot{x}(t) = Ax(t) \tag{13}$$

Where *x(t)*, the state vector, is given by

$$x = [\Delta\delta \quad \Delta\omega \quad \Delta E_q^{'} \quad \Delta E_{fd}]^T \tag{14}$$

The system matrix *A* is given by

$$A = \begin{bmatrix} 0 & \omega_0 & 0 & 0 \\ \frac{\text{-}K_1}{M} & 0 & \frac{\text{-}K_2}{M} & 0 \\ \frac{\text{-}K_4}{T_{do}^{'}} & 0 & \frac{\text{-}1}{K_3 T_{do}^{'}} & \frac{1}{T_{do}^{'}} \\ \frac{-K_e K_5}{T_e} & 0 & \frac{\text{-}K_e K_6}{T_e} & \frac{\text{-}1}{T_e} \end{bmatrix} \tag{15}$$

By varying *P* and/or *Q* to cover a wide range of system loading, the parameters K_1 to K_6 are computed. Then for every *P* and *Q* combination, the eigenvalues of the system are calculated.

B Stabilized system

B.1. With the power stabilizing signal activated the order of the system increases to six. In this case

$$u = \frac{K_1\Delta\delta + K_2\Delta E_q^{'}}{(S + \frac{1}{T})^2} \tag{16}$$

The state vector is given by

$$x = [\Delta\delta \quad \Delta\omega \quad \Delta E_q^{'} \quad \Delta v_1 \quad \Delta v_2]^T \tag{17}$$

Where Δv_1 and Δv_2 are auxiliary state variables. The closed-loop system matrix is given by equation (18). To stabilize the system over all changes of loading, the genetic algorithm is used. It is called genitor algorithm [4]. To calculate the objective function as given by equation 13, the eigenvalues of the system matrix *A* are computed for a selected set of grid points in the real-power/reactive power domain for each of the members of the current population. The values of the objective functions thus obtained are fed to the GA in order to produce the next generation of chromosomes. The procedure is repeated until the population has converged to some minimum value of objective function producing the optimal parameter set. The following GA parameters were used in this case: Population size=100, Length of each chromosome=48, Maximum number of generation=320, Crossover probability: 0.9, Mutation probability: 0.001. The optimum values of the PSS parameters were found to be K=7.4712, T=0.3104, J=-0.4137. These values of K and T ensure that eigenvalues corresponding to the operating points, are located in the left-hand side of the complex s-plane for the entire loading range, in fact to the left of the line S=-0.4137 as evident from the value of the objective function.

$$A=\begin{bmatrix} 0 & \omega_0 & 0 & 0 & 0 & 0 \\ -\frac{K_1}{M} & 0 & -\frac{K_2}{M} & 0 & 0 & 0 \\ -\frac{K_4}{T'_{do}} & 0 & -\frac{1}{K_3T'_{do}} & \frac{1}{T'_{do}} & 0 & 0 \\ -\frac{K_eK_5}{T_e} & 0 & \frac{K_eK_6}{T_e} & -\frac{1}{T_e} & 0 & -\frac{K_e}{T_e} \\ K_1K & 0 & K_2K & 0 & -\frac{1}{T} & 0 \\ K_1K & 0 & K_2K & 0 & -\frac{1}{T} & -\frac{1}{T} \end{bmatrix} \tag{18}$$

B.2. With the speed stabilizer signal the order of the system becomes 6 again. Here:

$$u=K_c\frac{ST}{1+ST}\frac{(1+ST_1)}{(1+ST_2)}\Delta\omega \tag{19}$$

The state vector is given by:

$$\dot{x}(t)=Ax(t),\ x=[\Delta\omega \quad \Delta\delta \quad \Delta\mathrm{E}_{\mathrm{fd}} \quad \Delta\mathrm{v}_1 \quad \Delta\mathrm{v}_2]^{\mathrm{T}} \tag{20}$$

$$A=\begin{bmatrix} 0 & \omega_0 & 0 & 0 & 0 & 0 \\ -\frac{K_1}{M} & 0 & -\frac{K_2}{M} & 0 & 0 & 0 \\ -\frac{K_4}{T'_{do}} & 0 & -\frac{1}{K_3T'_{do}} & -\frac{1}{T'_{do}} & 0 & 0 \\ -\frac{K_eK_5}{T_e} & 0 & -\frac{K_eK_6}{T_e} & -\frac{1}{T_e} & 0 & \frac{K_e}{T_e} \\ -\frac{K_1}{M} & 0 & -\frac{K_2}{M} & 0 & -\frac{1}{T} & 0 \\ \frac{K_cK_1T_1}{MT_1} & 0 & -\frac{K_cK_2T_2}{MT_2} & 0 & \frac{K_c\left(1-\frac{T_1}{T}\right)}{T_2} & -\frac{1}{T_2} \end{bmatrix} \tag{21}$$

It is worth noting that the sign of speed change is vice versa in relation to the sign of power change, so the output of speed PSS applied to the AVR reference must be opposite to power PSS. The following GA parameters were used in this case: Population size=150, Length of each chromosome=48, Maximum number of generation=350, Crossover probability: 0.96, Mutation probability: 0.001. The optimum values of the PSS parameters were found to be K_c=13.236, T_1=2.134, T_2=0.032; J=-0.5423. In order to get view of two PSS performances and effect of them some time domain simulations were done, it was tried to check the PSS performance under some critical points such as heavy duty, light duty and lead duty. Three operating points were chosen: (P=1pu, Q=-0.2pu), (P=0.1pu, Q=-0.2pu), (P=1pu, Q=0.8pu).The first point is the worse operating point of system. Tables 1, 2 and 3 show the mechanical eigenvalues of the system at different operating points with and without PSS's.

Table 1. Eigenvalues of system at P0=1pu, Q0=0.2pu, Xe=0.2pu

DEFENITION	WORST CASE EIG
Without PSS	0.0484+j7.8331
With power PSS	-0.4237+j13.2457
With speed PSS	-0.9216+j1.1256

Table 2. Eigenvalues of system at P0=0.1pu, Q0=-0.2pu, Xe=.2pu

DEFENITION	WORST CASE EIG
Without PSS	-0.1759+j4.1871
With power PSS	-0.6338+j2.9514
With speed PSS	-0.8934+j2.1056

Table 3. Eigenvalues of system at P0=1pu, Q0=0.8pu, Xe=0.2pu

DEFENITION	WORST CASE EIG
Without PSS	0.0120+j7.3691
With power PSS	-1.5427+j11.1238
With speed PSS	-1.6738+j2.3547

For test of system at first and second operating points an impulse input of reference torque by magnitude of .05pu and for third operating point an impulse input of reference voltage by magnitude of 0.1pu are applied to system as disturbances. Simulation results are shown in figures 3 to 5. It happens many times in power systems that after occurring a fault, a tie line cuts off by reclosures, this weaks the system stability and starts low frequency oscillations. During the tuning process of PSS parameters the case of cutting off one tie line was considered, Here the PSS performance is shown by time domain simulation, the operating point is chosen as (P=0.9pu; Q=-0.1pu) and it is considered that a fault occurs in the system and then clears by cutting off a tieline. Figure 6 shows the system after clearing the fault. The simulation results in this case are shown in figure 7. It is worth noting that the suggested technique can be applied to stabilize a multimachine system. It only differs from the single machine-infinite bus case in the amount and time of computation. Meanwhile, the PSS design can be achieved using the suggested technique by considering one operating point only, i.e. N=1. To test this idea on multimachine systems and one operating point, a three machine system is considered which is shown in figure 8. Using Heffron-Philips model for multimachine systems, the K_1-K_6 matrixes are computed and the model analysis is done. Table 4 shows the mechanical eigenvalues of the system without any PSS. It is clear that eigenvalues 5 and 6 have a very poor damping. To give a sufficient damping to this system, it is tried to tune the parameters of 3 power input stabilizers each mounted on one generator by help of genetic algorithm, for each PSS 2 parameters must be tuned. The following GA parameters were used in this case:

Population size=150, Length of each chromosome=48, Maximum number of generation=350, Crossover probability: 1.0, Mutation probability: 0.003.

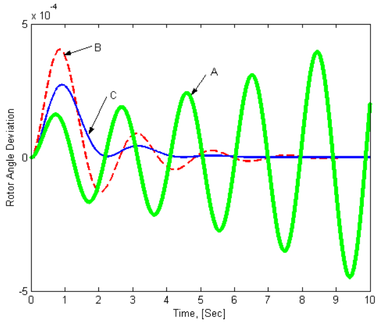


Fig. 3. Rotor angle deviation of system at P=1pu, Q=-0.2pu, X_e=0.2pu, A: without PSS, B: with power input PSS, C: with speed input PSS

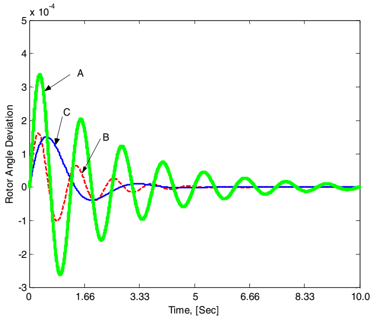


Fig. 4. Rotor angle deviation of system at P=0.1pu, Q=-0.2pu, X_e=0.2pu, A: without PSS, B: with power input PSS, C: with speed input PSS

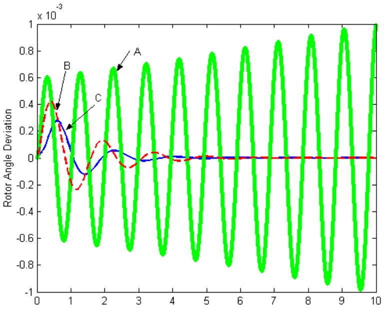


Fig. 5. Rotor angle deviation of system at P=1pu, Q=0.8pu, X_e=0.2pu, A: without PSS, B: with power input PSS, C: with speed input PSS PSS

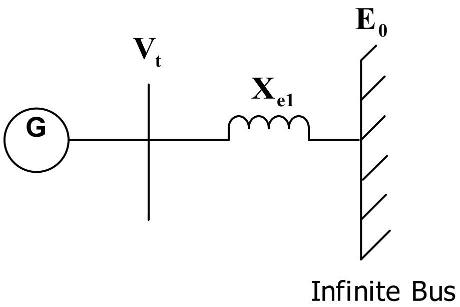


Fig. 6. System after cut off one tieline

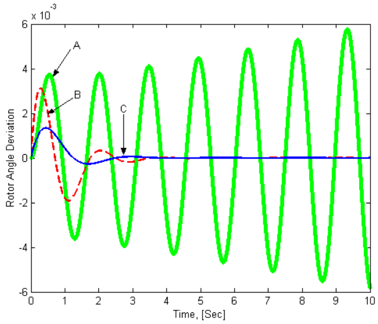


Fig. 7. Rotor angle deviation of system at P=0.9pu,Q=-0.1pu, X_e=0.4pu, A: without PSS, B: with power input PSS, C: with speed input PSS

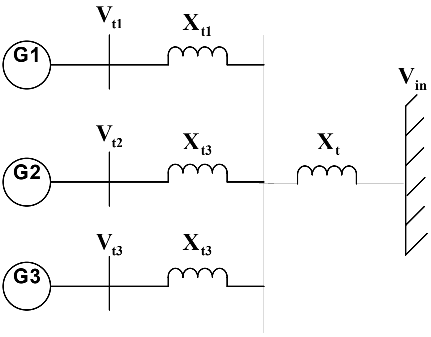


Fig. 8. A three machine test system

Table 4. Mechanical eigenvalues of system without PSS design

Number	Mechanical eigenvalues of system without PSS's
1,2	-1.9867+j12.3417;-1.9867-j12.3417
3,4	-4.6712+j8.9615; -4.6712-j8.9615
5,6	-0.1012+j9.0142; -0.1012-j9.0142

Table 5 shows the eigenvalues of system after tuning and installing of PSS's. The simulation results are shown in figure 9.

Table 5. Mechanical eigenvalues of system without PSS design

Number	Mechanical eigenvalues of system after installation of PSS's
1,2	-0.7218+j3.1467,-0.7218- j3.1467
3,4	-0.7632+j5.8716, -0.7632-j5.8716
5,6	-0.9114+j2.7154, -0.9114-j2.7154

The suggested technique not only tunes the parameters of PSSs but also finds the optimum location for mounting of PSSs simultaneously and is more effective than SPE method [5,6] which depends on operating point and considered placement of PSSs and tuning of parameters individually.

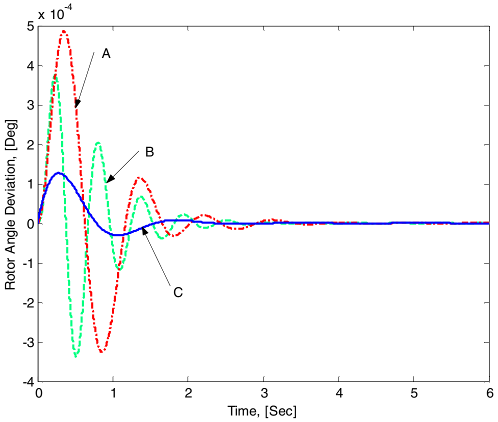


Fig. 9. Rotor angle deviation of three machine system in existence of PSS’s, A: Rotor angle of machine 1, B: rotor angle of machine 2, C: rotor angle of machine 3

6 Conclusion

The coordinated placement and tuning of decentralized power system stabilizers over a wide range of operating condition was investigated. The power system operating at various loading is treated as a finite set of plants .The problem of selecting the parameters of a PSS which stabilizes this set of plants has been converted to a simple optimization problem solved by GA and an eigenvalue based objective function. A single machine-infinite bus system demonstrated the suggested technique. It was shown that it is possible to select a single set of the PSS parameters to ensure the stabilization of the system for the entire loading range. The suggested technique was also applied on a multi-machine system, the results of time domain simulation showed that the designed PSSs have good performance in the system and work coordinately. It is clear that if the results of genetic algorithm lead to a very small K_i (the gain of PSS for ith machine) it means that the PSS shall not be mounted on ith generator, and simultaneous placement and tuning of power system stabilizers is achieved. For the speed input PSS, at least *m* parameters must be optimized more than power input PSS and the computation time increases in this case and the optimization problem may not converge. By analyzing behavior of system when different PSSs exist, it is understood that either power PSS or speed PSS have special advantages and disadvantages so the use of a combined PSS using both power and speed signals as input is recommended.

Reference

[1] A.B.Adbennour and K.Lee, "A Decentralized Controller Design For a Power Plant Using Robust Local Controllers And Functional Mapping", IEEE T-Energy Conversion, Vol.11,No.2, PP 394400June 1996.

[2] D.E.Goldberg, "Genetic Algorithm in Search, Optimisation And Machine Learning", Addison-Wesely, Reading MA , 1989

[3] D. E. Goldberg, “Genetic Algorithms in Search optimization and Machine Learning”, Reading, MA: Assison-Wesley Publishing Company, Inc, 1991

[4] E.Z.Zhou, O.P.Malik and G.S.Hope, "Theory And Method For Selection Of Power System Stabiliser Location", IEEE T-Energy Conversion, Vol.6, No 1, PP 101-109, March 1991

[5] E.Z.Zhou, O.P.Malik and G.S.Hope, "Design Of Stabiliser For A multimachine Power System Based On Sensitivity Of PSS Effect", IEEE T-Energy Conversion Vol.6,No1 pp606-612, March 1992

[6] F.P.Demello, L.N.Hannett and J.M.Undrill, "Practical Approach To Supplementary Stabilising From Accelerating Power", IEEE Transaction, 1978, PAS97, PP 1515, 1522

[7] JJ.Grefenstette, "Optimisation Of Control Parameters For Genetic Algorithm", IEEE Trans, 1986, SMC16,(1),PP 122-128

[8] W.E.Schmitendorf, R.W.Benson, O.shaw and S.Forrest, "Using Genetic Algorithm For Controller Design" Proceedings Of AIAA Conference on the Guidance, Navigation and Control, Hilton Head Island, South Carolina , August 1992,PP 757-761.

[9] Y.N.Yu "Electric Power System Dynamics", Academic Press, 1983

An Application of Adaptive Genetic Algorithm in Financial Knapsack Problem

Kwok Yip Szeto and Man Hon Lo

Department of Physics, Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, SAR, China
phszeto@ust.hk

Abstract. We apply adaptive genetic algorithm to the financial knapsack problem, which aims at maximizing the profit from investment with limited capital. Since the performance of genetic algorithms is critically determined by the architecture and parameters involved in the evolution process, an adaptive control is implemented on the parameter governing the relative percentage of preserved (survived) individuals and reproduced individuals (offspring). The portion of preserved individuals is kept to a proportion to the difference between the fitness of the best and average values of individuals in the population. Numerical experiments on knapsack problems with N ($150 \le N \le 300$) items are analyzed using the mean-absolute deviation generations against the median first passage generations to solutions. Results show strong evidence that our adaptive genetic algorithm can achieve the Markowitz investment frontier: the risk of missing the global optimum can be minimized by reducing the persevered popolution with increasing difficulty of the problem.

1 Introduction

Genetic algorithm (GA) is a technique of evolution programming [1-3] based on the Darwinian principle of survival of the fitness. Through simple encoding schemes to represent individuals in a population of potential solutions, complex phenomena are described by the evolution of these simple units. The flexibility of genetic algorithms attracts many usages in tackling science and engineering problems such as time series forecasting [4], pattern recognition [5], and cryptography [6.7]. Here we demonstrate the application of adaptive genetic algorithms in solving zero-one Knapsack problem in finance. In particular, the concept of investment frontier of Markowitz [8,9] is incorporated in locating the best parameterization

In GA, individuals with high fitness are always preserved, while the unfit ones are replaced by new individuals in the next generation. The idea is that only the best group of potential solutions survives and they can take part in reproducing offspring with higher fitness. These steps often ensure a monotonic increase of the total fitness of populations. In the knapsack problem, it is a searching process under restrictions. Therefore, it is necessary to avoid producing some invalid or illegal individuals, which violate the restrictions such as the overuse of the budget. In order to maximize

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1220-1228, 2004.

the fitness of individuals under restrictions, crossover and mutation operators are arranged in a special way that to rectify illegal individuals to vaild ones.

In adaptive GA, we like to design an automatic mechanism for the selection of parameters, so that the specific problem under investigation tells us the optimal set at a particular generation. Thus, we will not fix the rates for survival, crossover and mutation. However, we will introduce certain rules and restrictions to the architecture so that the system can adjust these rates in order to provide an optimal performance. In this investigation, we keep the preserved proportion (α) in our adaptive genetic algorithm to be a constant, and demonstrate the dependence of the performance on α.

2 Methodology

Only zero-one knapsack problem is discussed in this paper. The aim of using genetic algorithm is to evaluate a strategy to maximize the profit from the function of the knapsack problem.

2.1 Zero-One Knapsack Problem

The mathematical function of knapsack problem is as follow:

$$\text{Maximize:} \quad \sum_{i=0}^{N-1} p_i s_i \tag{1}$$

$$\text{Restriction:} \quad \sum_{i=0}^{N-1} w_i s_i \le C \tag{2}$$

p_i is the expected profit of the i-th item, which is a number randomly generated in the range [1, 10000].

w_i is the weight of the i-th item, which is a number randomly generated in the range [1, 1000].

s_i is an integer either 1 or 0 corresponding to the i-th item.

N is the total number of items in the knapsack problem.

C is the capital, which is a constant somewhere in-between zero and the sum of $\{w_i\}$. In our convection, we approimate C equal to the sum of products of expected values of $\{w_i\}$and $\{s_i\}$, i.e,

$$C = \sum_{i=0}^{N-1} \overline{w_i} \cdot \overline{s_i}$$

$$\Rightarrow \quad C = N \cdot (500.5)(0.5) \approx 250N \tag{3}$$

Our task is to find a set of $\{s_i\}$ that is simultaneously consistent with Eqs. (1) and (2). It can be seen that the total number of possible solutions is 2^N, which means that the difficulty of exhaustive search for solution increases exponentially with N.

2.2 Encoding Scheme and Fitness

In genetic algorithm, a chromosome is encoded as a binary string $\{s_i\}$, with the measure of its fitness f by :

$$f = \sum_{i=0}^{N-1} p_i s_i \tag{4}$$

A population of chromosomes is created as the set of strings with corresponding set of fitness $\{f_i\}$. Individuals with high fitness is desirable to the Eq.(1), but may not satisfy Eq.2, such as the set $\{s_i=1\}$. In order to produce a set of legal individuals with high fitness, we need to use genetic operators such as crossover and mutation.

2.3 Crossover Operator

Crossover is used to reproduce offspring by choosing two subsequences of genes from two individuals (or parents) separately, and swapping such subsequences without altering the order and position of other genes. A subsequence of genes is selected by two random cut points, which serve as boundaries for the swap. For example, assuming $N = 10$:

Parent 1: $\{s_i\}$= (1101101001) ➔ (1101001101)

Parent 2: $\{s_j\}$= (0111001111) ➔ (0111101011)

However, if the two individuals are initially identical, this crossover will make no difference. In this case, we allow only one of them to survive, and regenerate the other one by a random binary string.

2.4 Mutation Operator

After crossover, individuals generally have a different fitness, which may not satisfy Eq.2. We then use mutation on these *"illegal"* individuals until they satisfy Eq.2. It applies to an individual under the two criteria:

1) If $\sum_{i=0}^{N-1} w_i s_i < C$, it randomly filps a position in $\{s_i\}$ that is zero to one repeatly before the inequility cannot be hold.
2) If $\sum_{i=0}^{N-1} w_i s_i > C$, it randomly filps a position in $\{s_i\}$ that is one to zero repeatly until the inequility cannot be hold.

With such corrections, all individuals in the population will satisfy Eq.2.

2.5 Average-Selection Method

The importance of Darwinian evolution in the context of our optimization problem lies in the selection of survivors with high fitness. However, the fitness threshold for survival is quite arbitrary. To make a less biased decision, we employ the *average* of the fitness distribution as the threshold. Statistically, an individual with fitness somewhere above the average is considered as good and some of them are allowed to survive to the next generation. This survival threshold is assumed to be a portion of the difference between the highest fitness and the average fitness. We thus introduce the following parameter,

$$\Psi = f_{average} + \alpha \cdot \left(f_{highest} - f_{average} \right) \tag{5}$$

where α is a proportional constant in the range [0, 1].

In order to increase the diversity of the survival group, we only pick those individuals with distinguishable *fitness* values higher than Ψ as survivors.

2.6 The Adaptive Architecture

By using the parameter Ψ, the adaptive structure of the average-selection method is constructed as follows:

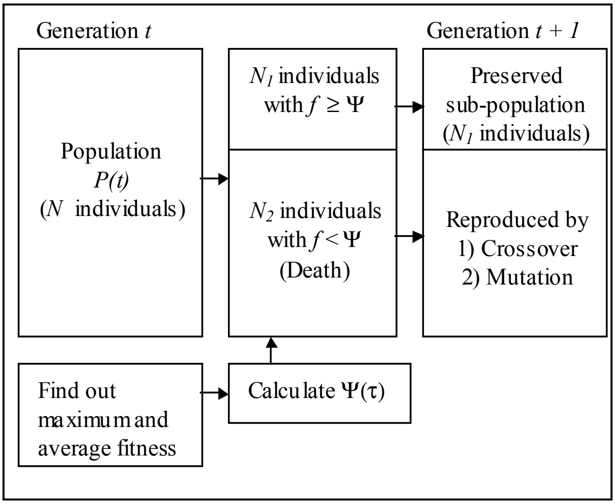

Generation t
Generation $t + 1$
Population $P(t)$ (N individuals)
N_1 individuals with $f \geq \Psi$
N_2 individuals with $f < \Psi$ (Death)
Preserved sub-population (N_1 individuals)
Reproduced by 1) Crossover 2) Mutation
Find out maximum and average fitness
Calculate $\Psi(\tau)$

Fig.1. Adaptive architecture of the average-selection method.

Since searching process is computationally intensive, our Average-Selection method is specifically designed to minimize the computation involved in tuning the adaptive parameter in genetic algorithm.

3 Experiments and Results

Knapsack problems with size *N* equal to 150, 200, 250 and 300 are solved with our adaptive genetic algorithm. In each problem, we also investigate different proportional constant α, starting from 0.1 to 0.9, with an increment of 0.1. We also fix the number of chromosomes at 100. We run each experiment 100 times and for each experiment, we record the number of generations required to locate the global maximum. We also analyze the data using median first passage generation and mean-absolute deviation of generation to the solution.

3.1 Median First Passage Generation

In the searching process using genetic algorithm, we only record the number of generations that the solution (global maximum) first appears. This is called the first passage generation. Such experiments are done 100 times and we can sort out the median of the first passage generation to the solution. The merit of using median here is to filter out any extreme performance that dominates other samples.

3.2 Mean-Absolute Deviation of Generation

The mean-absolute deviation of the distribution of first passage generation to solution can be interpreted as the risk to find the global maximum (solution) for given number of generations.

3.3 Performance of the Genetic Algorithm

The performances of the genetic algorithm in each problem are listed below, with the constant α increases from 0.1 to 0.9 in the trend of clockwise represented in the following figures.

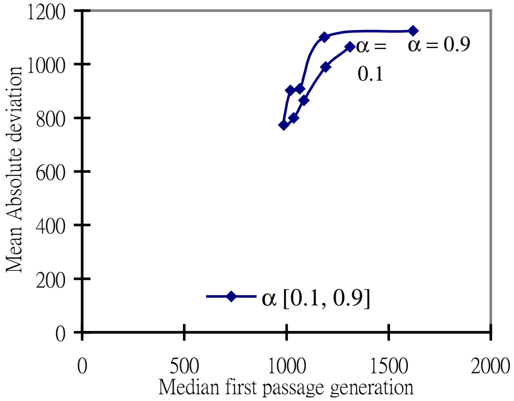


Fig.2. The performance of GA in a knapsack problem with 150 items.

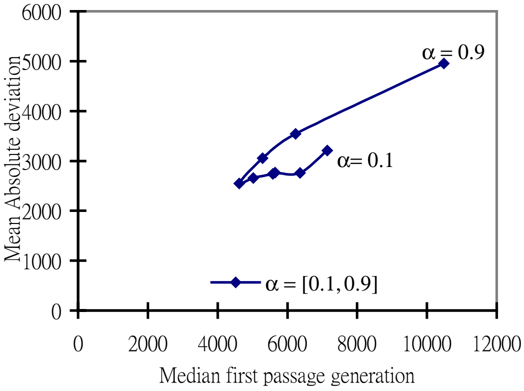


Fig. 3. The performance of GA in a knapsack problem with 200 items.

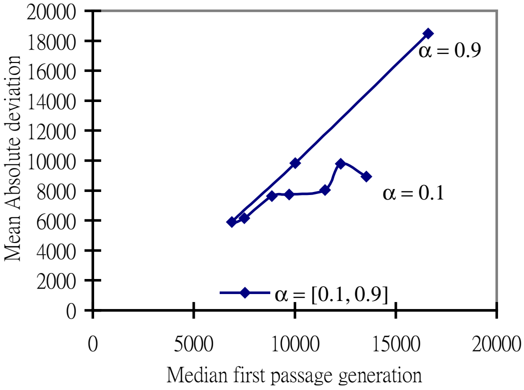


Fig. 4. The performance of GA in a knapsack problem with 250 items.

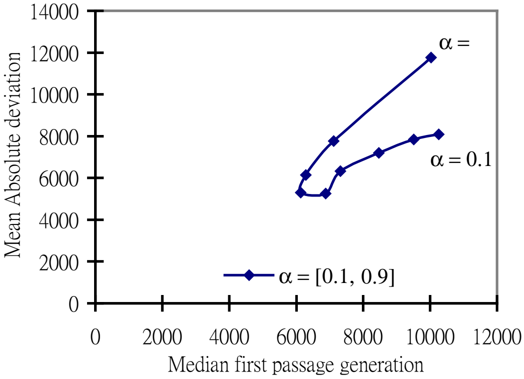


Fig. 5. The performance of GA in a knapsack problem with 300 items.

From Fig.2-5, we see that different settings of α will give different risk and median first passage generation. In fact, these figures are realization of the investment yield frontier of Markowitz [8,9]. Ideally, one should choose an α that give the point on the investment frontier that is closest to the origin. This choice of α defines the optimal settings in the context of our adaptive genetic algorithm. They are summarized below.

Table 1. Optimial settings of α in each knapsack problem.

Number of items	Optimal setting of α	Median first passage generation	Mean-absolute deviation of generation
150	0.5	987	773
200	0.6	4622.5	2546
250	0.7	6889.5	5894
300	0.6	6134	5300

4 Discussion

In searching for solution with genetic algorithm, we like to find the global solution with greatest confidence and with the shortest time. However, as shown by Huberman et al, [9], it is not easy to minimize the searching time and its risk (or deviation) simultaneously. In general, we need to find an organizational framework to locate the *"investment frontier"* [8] for computational resource. In this paper, we describe in the context of adaptive parallel genetic algorithm a general framework to fulfill such criteria. Our experiments demonstrate that in solving the financial knapsack problem, the proportional constant α is critical. This α parameter can be used to locate the optimal setting of crossover, mutations and other genetic operator. From the Table 1, we postulate that the optimal setting of α varies with the difficulty of the problem. We see that the optimal time to solution (median first passage generation) with corresponding deviation rises with the optimal α. Since a good measure of the difficulty of a problem is the time needed for its solution, the results in Table 1 reveal that the one should preserve a small amount of individuals (large α) in each generation in which the difficulty of the problem is high, and vice versa (small α) for an easier problem. Usually, the difficulty of finding the solution increases with the number of items in the problem, as the total number of combination is equal to 2^N, which will certainly produce a drop in the probability to the solution. However, this is not the only factor in determining the difficulty of the problem. Since the various weights in our knapsack problems are randomly generated, it is possible to have traps near the global maximum that increases the difficulty of the problem. In Table 1, we see an increasing trend in the difficulty of problems with items from 150 to 300. However, the difficulty of the problem of 250 items is larger than that of 300 items. This is because the problem of 250 items that we generated contains more traps than that of the 300 items. In the context of physics, a system with more traps generally is more complex and the difficulty of finding the global optimum increases. One factor that determines the number of traps is the number of items, but it is not the unique factor. The distribution of weights will also affect the number of traps. Indeed, it is the distribution of traps near the global optimum that is more difficult to handle. We may interpret the numerical result in Table 1, where the difficulty of the 300 item knapsack is lower than that of the 250 items knapsack problem as an indication of the importance of the distribution of weights w_i and probabilities p_i on the nature and number of traps, though we have not analyzed the traps in each case. Nevertheless, supposing that the number of traps and the nature of traps together determine the overall difficulty of a problem, then for systems with many traps for local optimum, the search for global optimum requires keeping less of those individuals with high fitness, so that there is more room for the reproduction of new individuals, resulting in a more diversified population that may leave a local trap more easily. This is our heuristic argument for a larger α (keeping smaller number of fit survivors) for a more difficult problem. Knowing the significance of the parameter control in α, one can apply our adaptive genetic algorithms in the practical situation. As the difficulty of each new knapsack problem is not known, one can control the α by supervising the rate of change of the maximum fitness, which reveals the progress of the search at that particular moment. The setting of α should be increased gradually with respect to the decreasing of the rate of change of the maximum fitness, or vice versa. The further application of our adaptive genetic algorithm is to handle the time-dependent knapsack problem, which contains investment items with their values changing from time to time. The values of each items are updated at a given time interval while the genetic algorithm co-evolve with the problem. We expect that this will lead to suitable solution for real time situation. We are in the process of doing optimization for time-dependent knapsack problems using adaptive parallel genetic algorithms, with possible application in stock market and computer network.

5 Conclusion

We applied an adaptive genetic algorithm to solve the knapsack problem. We find that the mean time to solution and the risk of finding a solution depend on the size of the preserved proportion and the difficulty of the problem. It is observed that there always exists an optimal setting of parameter that can minimize the searching time and its deviation simultaneously. In order word, within the limited examples for the knapsack problems that we have tried, we find the existence of a parameter that allows us to determine the investment frontier for computational resource. Though it is generally hard to accurately measure the difficulty of the specific problem, the conventional wisdom is that the more items a knapsack problem contains, the higher degree of its difficulty. In this simplistic view of the knapsack problem, a general guideline emerges from our studies : keep the preserved proportion in a relatively low value for problem with large number of items. This work is supported by HKUST6144/00P and HKUST6157/01P grant.

References

1. Holland, J.H., *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.
2. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
3. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
4. Szeto,K.Y., and Cheung,K.H. "Multiple time series prediction using genetic algorithms optimizer," *Proceedings of the International Symposium on Intelligent Data Engineering and Learning*, Hong Kong, IDEAL'98, 127-133, 1998.
5. Fong, L.Y. and Szeto, K.Y., "Rule extraction in short memory time series using genetic algorithms," *European Physics Journal B*, Vol.20, 569-572 (2001)
6. Li. S.P. and Szeto, K.Y., "Crytoarithmetic problem using parallel Genetic Algorithms," Mendl'99, Brno, Czech, 1999.
7. Lo,M.H., and Szeto,K.Y., Searching Solutions in the Crypto-arithmetic Problems: An Adaptive Parallel Genetic Algorithms Approach, Advances in Artificial Intelligence Eds. Yang Xiang and Brahim Chaib-draa, LNAI2671,AI2003, Springer-Verlag 2003. pp.81-95.
8. Markowitz, H., J. of Finance, Vol.7, 77, (1952)
9. Huberman,B.A., Lukose,R.M., and Hogg,T., An economic approach to hard computational problems, Science, Vol. 275, 51-54, (1997)

Assimilation Exchange Based Software Integration

L. Yang and B.F. Jones

Applied Computing Department, the Derbyshire Business School, University of Derby, Derby, DE22 1GB, UK
L.yang@derby.ac.uk

Abstract. This paper describes an approach of integrating software with a minimum risk using Genetic Algorithms (GA). The problem was initially proposed by the need of sharing common software components among various departments within a same organization. The main contribution of this study is that the software integration problem is formulated as a search problem and solved using a GA. A case study was based on an on-going software integration project carried out in the Derbyshire Fire Rescue Service, and is used to illustrate the application of the approach.

1 Introduction

To decrease costs, increase efficiency and service new requirements, organizations have spent considerable resources eliminating silos of information through application integration. There are a number of available approaches for achieving this integration. For example, point-to-point integration [7] involves establishing a basic data interchange infrastructure between each pair of applications. Systems are loosely coupled, permitting a degree of application independence. The shortcoming of the point-to-point integration is that the number of interfaces required grows exponentially. The impact of minor changes in communication requirements is significant. Maintenance is clearly a nightmare. Message bus based integration [1, 7] requires interfacing each application to the message bus through an adapter. Each application has only one programmatic interface, the message bus. Applications communicate by publishing a message to the bus, which delivers the message to those who require.

The above two approaches focus on the communication between different applications which are integrated. An Assimilation Exchange, AX, focuses on which part of the applications should be integrated in order to implement certain functions. An AX is an advanced exchange that assimilates parts of existing systems from various applications and uses them to create a value-adding infrastructure, shared by the partners. Each participant contributes to the AX a number of components of their information infrastructure, which are assimilated, shared and inter-operated within the AX [11]. The concept of the AX is illustrated in Fig. 1. From software engineering perspective, the AX is an integration of a number of software components offered by a number of participants to achieve a defined goal with a minimum risk. The challenge is how to choose these components from participants to achieve the defined

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1229-1238, 2004.

functionality and minimize the risk at the same time. The communication between these chosen components is important as well and has not been discussed in this paper due to the limitation of the space.

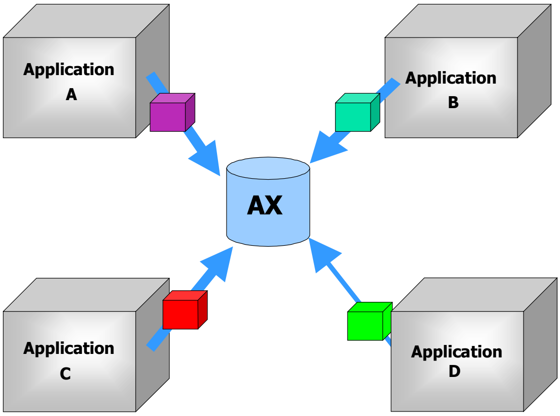


Fig. 1. Conceptual model of the AX

This paper focuses on the AX based software integration and proposed an approach of integrating software with a minimum risk using GAs. The main contribution of this study is that the software integration problem is formulated as a search problem and solved using a GA. This paper is organised as follows: Section 2 reformulates the AX based software integration as a search problem. Section 3 is the application of the approach to the Derbyshire Fire Rescue Service case study. This case study was based on an on-going software integration project. Section 4 concludes the paper and discusses future extensions.

2 Reformulating Software Integration as a Search Problem

In Harman and Jones□s work [3] software engineering has been, in general, reformulated as a search problem. In order to do so, it is necessary to define:

- a representation of the problem which is amenable to symbolic manipulation,
- a fitness function defined in terms of this representation, and
- a set of manipulation operators.

The representation of a candidate solution is critical to shaping the nature of the search problem. Floating-point numbers and binary code are representations, which are frequently used in existing applications. The fitness function is the characterization of what is considered to be a good solution. Generally it will be sufficient to know which of two candidate solutions is the better according the fitness function. Different search techniques use different operators. As a minimum requirement, it will be necessary to mutate an individual representation of a candidate solution. Genetic algorithms include three operators: mutation, crossover and reproduction. These issues are covered in detail in the general literature on meta-heuristic search [2, 9]. The areas of software engineering which meta-heuristic search can be applied focus on software testing and test data selection [4]. To the authors□ knowledge, meta-heuristic search has never been used in software integration before. The principal intention of this section is to demonstrate that the reformulation of software integration as a search problem is conceptually feasible.

2.1 Mathematical Model

As described in Sections 1, the AX is an integration of a number of software components offered by a number of participants. The selected AX components should achieve a defined functionality and minimise the risk as well.

In most cases it may be hard to define functionality as numerical values. The symbolic description and the set theory are used in this study for modeling of the functionality of the AX. Obviously, the overall required functionality, F, of the AX is the joint set of all the required sub-functionality, F_k, as shown in Equation (1), and must be provided by the aggregated functionality, f_{ij}, of all the selected components m_{ij}, as shown in Equation (2). $Occur_{ij}$ is a binary value, representing the occurrence of a component m_{ij} in the AX. K is the number of the sub-functionalities. M is the number of participants in the AX. N^i is the number of the components of the participant i $_{(i=1,\ \square\ M)}$.

$$F = \bigcup_{k=1}^{K} F_k \tag{1}$$

$$F = \bigcup_{i=1}^{M} \bigcup_{j=1}^{N^i} f_{ij} \times Occur_{ij} \tag{2}$$

Where

$$Occur_{ij} = \begin{cases} 1 & m_{ij} \quad present \quad (Selected) \\ 0 & m_{ij} \quad absent \quad (notSelected) \end{cases} \tag{3}$$

The functionality of the component f_{ij} can be the same as one of the sub-functionalities denoted by the symbol □=□, or include more than one sub-functionalities denoted by the symbol □⊃□, or be excluded from the overall required functionality, F, denoted by the symbol □∩□. They are described in the following Equations:

$$\left\{\begin{array}{c} f_{ij} = F_k \\ f_{ij} \supset F_k \\ f_{ij} \cap F_k = \phi \end{array}\right. \tag{4}$$

where □is an empty set.

Similarly, the overall risk, R, is contributed by all the selected components m_{ij}, and is represented in Equation 5.

$$R = \sum_{i=1}^{M} \sum_{j=1}^{N^i} R_{ij} \times Occur_{ij} \tag{5}$$

where R$_{ij}$ is the risk associated with the component m_{ij}, which is composed of a number of associated metrics or parameters, p_{ij}^{l} *(l=1, 2, ..., L)*, such as reliability, coupling, security, complexity, and so on.

$$R_{ij} = R_{ij}(p_{ij}^{1}, \cdots, p_{ij}^{l}, \cdots, p_{ij}^{L}) \tag{6}$$

Equations 1 to 6 form the mathematical model of the AX based software integration problem. This problem is formally described as: choose the values of *Occur$_{ij}$* in Equations 2 and 5 for the components m$_{ij}$ contributed by the participants {P$_1$, P$_2$, □ , P$_M$} so that the risk R described in Equation 5 achieves its minimum value and the overall functionality F described in Equation 1 fit Equation 2.

2.2 Reformulating as a GA Problem

The above problem could be solved using genetic algorithms by applying the constraint of Equation 2 and using the overall risk R as the fitness function. First all, a representation of the problem, i.e. a chromosome, is required. The chromosome could be expressed as a binary string, which is the set of the occurrences, Occur$_{ij}$, of all the possible components m$_{ij}$.

$$Occur_{11} \quad Occur_{12} \quad \cdots \quad Occur_{1N^1} \quad \cdots \quad \cdots \quad Occur_{M1} \quad Occur_{M2} \quad \cdots \quad Occur_{MN^M} \tag{7}$$

The string length is equal to the total number of the components m$_{ij,}$ $\sum_{i=1}^{M} N^i$, A population is composed of a number of chromosomes. The initial value of the population is formed randomly and then subjected to reproduction, mutation, and crossover to form the next population, which should contain better possibilities to minimize the fitness function. The overall risk representation *R* is chosen as the fitness function here.

$$f_{fitness} = \sum_{i=1}^{M}\sum_{j=1}^{N^i} R_{ij} \times Occur_{ij} \tag{8}$$

In order to satisfy the constraint of Equation 2, a penalty term is added into the fitness function. This penalty term is represented by the number of missing sub-functionalities E.

$$f_{fitness} = \sum_{i=1}^{M}\sum_{j=1}^{N^i} R_{ij} \times Occur_{ij} + E \tag{9}$$

The number of missing sub-functionalities, E, can be found by comparing the joint set of $F_{k\ (k=1,\ 2,\ \square\ ,\ K)}$ and the joint set of f_{ij}, i.e.

$$\bigcup_{k=1}^{K} F_k - \bigcup_{i=1}^{M}\bigcup_{j=1}^{N^i} f_{ij} \times Occur_{ij} \tag{10}$$

Where K is the total number of sub-functionalities, which is obtained from the decomposition of the AX overall functionality. If all sub-functionalities have been implemented E will be equal to zero and the fitness function, $f_{fitness,}$becomes the value of the risk.

3 Software Component Risk Assessments

According to NASA Technical Standard [5], risk is a function of the possible frequency of occurrence of an undesired event, the potential severity of resulting consequences, and the uncertainties associated with the frequencies and severity. In the most risk assessment risk is defined as a combination of two factors: frequency (or possibility) of malfunctioning (failure) and the consequence of malfunctioning (severity), as shown in Equation 11. In large hierarchical systems, a system is composed of several subsystems, which in turn, are composed of components. The system risk is an aggregate of individual component risk factors [6, 10].

$$\text{Risk} = \text{frequency} \times \text{severity} \tag{11}$$

3.1 Frequency of Failure

The frequency of failure depends on the probability of existence of a fault combined with the possibility of exercising that fault. For the sake of the simplicity, in this study, we compute the risk by estimating the frequency of failure for each line code multiplied by the number of lines of code (LOC). Assuming that the frequency of failure for each line code is 10^{-5} per year, the frequency of failure for a 1000 lines code component is computed to be about 0.01 per year. For the companies who are using shrink-wrapped software in which the LOC is not known,.other proper risk analysis method is required.

3.2 Severity Analysis

Severity analysis is a procedure by which each potential failure mode is ranked according to the consequences of that failure mode. Severity considers the worst-case consequences of a failure determined by the degree of injury, property damage, system damage, and mission loss that could ultimately occur. The domain expert determines a severity for the faulty component for each scenario by comparing the faulty result with the normal operation. Severity classifications recommended by MIL_STD_1629A [8] are:

- Catastrophic: A failure may cause death or total system loss. The severity index is chosen as 0.95.
- Critical: A failure may cause severe injury, major property damage, and major system damage. The severity index is chosen as 0.75.
- Marginal: A failure may cause minor injury, minor property damage, and minor system damage. The severity index is chosen as 0.50.
- Minor: A failure is not serious enough to cause injury, property damage, or system damage, but will result in unscheduled maintenance or repair. The severity index is chosen as 0.25.

4 Case Study

Derbyshire Fire and Rescue Service (DFRS) has been selected as a case study to discuss the application of the proposed integration approach. There are eight physically independent systems being used in the DFRS. The goal of the case study is to deliver a single virtual application through the AX based software integration so that the common used components in these isolated systems can be shared and inter-operated within this virtual application. These eight systems are

- Mobilising System (MOB): provide the current locations of the available fire engines at every fire station in the DFRS, and record the emergency call details.
- Management Information System (MIS): provide the access to the fire incident databases, the personnel databases, and the relevant documents.
- Risk Assessment System (RISK): provide a building risk categorization and an access plan to higher risk premises.
- Geographical Information System (GIS): provide the risk information of buildings and areas in a visual way.

- Fire Safety System (SAFETY): is a database system, particularly designed for producing statistics reports.
- Crime and Disorder System (CRIME): store all the crime and disorder information, such as malicious call, hoax fire call, and vehicle crime.
- System (HYDRANT): provide hydrant information.
- Location Optimisation System (OPTIM): optimising fire station locations.

4.1 Mathematical Model of the DFRS

In terms of the above system description, there will be 8 participants in the AX. Each of them has a number of public components that might be contributed to the AX. The functionality of each component is represented in a set of symbolic variables. The risk generated by the components is computed in the form of Equation 11, and is represented in a numerical value. Table 1 lists the desired functions, which the AX is expected to offer. Table 2 summarizes the individual participants, the components provided by the participants, the functionality of each component, and the risk generated by them. Table 3 shows the decomposition of the functionality of each component. The risk in Table 2 is computed by the frequency of failure multiplied by the severity index for each component. The parameters of the mathematical model of the DFRS described in Equations 1 to 6 are list in Tables 1 to 3. Equation 6 has been simplified using Equation 11.

Table 1. The AX functions

Function (K=11)	Description
F_1	Provide the available resource information in the DFRS, including available fire engines and fire fighters.
F_2	Provide the current fire incident information.
F_3	Provide the fire incident information during different periods of time
F_4	Provide the fire risk categorization for a particular building
F_5	Provide an access plan for any particular higher risk premise
F_6	Provide the fire risk categorization for a particular area
F_7	Provide a forecasting function of the fire incident occurrence
F_8	Provide various crime and disorder information such as hoax fire call, malicious call.
F_9	Provide the location information of hydrant points
F_{10}	Provide the maintenance information of hydrant points
F_{11}	Provide a computing environment for locating fire stations, fire fighters and fire engines.

Table 2. Components and risks

Participant (M=8)	**Public components**	**Functionality**	**Risk** (R_{ij}, i=1, □ , 8; j=1, □ , N^i)	**Number of the components** (N^i, i=1, 2,..8)
P_1, MOB	m_{11}, m_{12}, m_{13}, m_{14}, m_{15}	f_{11}, f_{12}, f_{13}, f_{14}, f_{15}	0.375, 0.1875, 0.375, 0.375, 0.3	5
P_2, MIS	m_{21}, m_{22}, m_{23}	f_{21}, f_{22}, $f_{23,}$	0.025, 0, 0.0625	3
P_3, RISK	m_{31}, m_{32}, m_{33}	f_{31}, f_{32}, f_{33}	0.375, 0.375, 0.375	3
P_4, GIS	m_{41}, m_{42}, m_{43}	f_{41}, f_{42}, f_{43}	0.375, 0.3, 0.3	3
P_5, SAFETY	m_{51}, m_{52}	f_{51}, f_{52}	0.1, 0.25	2
P_6, CRIME	m_{61}, m_{62}	f_{61}, f_{62}	0.15, 0.15	2
P_7, HYDRANT	m_{71}	f_{71}	0.1	1
P_8, OPTI M	m_{81}	f_{81}	0.1	1

Table 3. Functionality of components

Function	**Description**	**Decomposition**
f_{11}	Provide the current available resource information in the DFRS, including available fire engines and fire fighters.	$f_{11}= F_1$
f_{12}	Provide the fire incident information during the latest eight hours.	$f_{12}\supset F_2$
f_{13}	Provide the fire risk categorization for a particular building.	$f_{13}= F_4$
f_{14}	Provide an access plan for any particular higher risk premise.	$f_{14}= F_5$
f_{15}	Provide the location information of hydrant points.	$f_{15}= F_9$
f_{21}	Provide the access to the fire incident database	$f_{21}\supset F_3$ $f_{21}\supset F_8$
f_{22}	Provide the access to the personnel database	$f_{22}\cap F_k=\Phi$, $_{k=1,□,11}$
f_{23}	Provide the access to the relevant documents	$f_{23}\supset F_{10}$
f_{31}	Provide the fire risk categorization for a particular building	$f_{31}= F_4$
f_{32}	Provide an access plan for a particular higher risk premise	$f_{32}= F_5$
f_{33}	Provide the fire risk categorization for a particular area	$f_{33}= F_6$
f_{41}	Provide the risk information of buildings	$f_{41}\supset F_4$
f_{42}	Provide the risk information of areas	$f_{42}\supset F_5$ $f_{42}\supset F_6$
f_{43}	Provide the location information of hydrant points	$f_{43}= F_9$
f_{51}	Produce statistics reports	$f_{51}\cap F_k=\Phi$, $_{k=1,□,11}$
f_{52}	Provide a forecasting function of the fire incident occurrence	$f_{52}= F_7$
f_{61}	Provide various crime and disorder information such as hoax fire call, malicious call.	$f_{61}= F_8$
f_{62}	Identify the higher crime and disorder areas	$f_{62}\cap F_k=\Phi$, $_{k=1,□,11}$
f_{71}	Provide the location and maintenance information of hydrant points	$f_{71}\supset F_9$
f_{81}	Provide a computing environment for locating fire stations, fire fighters and fire engines.	$f_{81}= F_{11}$

4.2 Results and Analysis

After introducing the parameters in Tables 1, 2 and 3 into the mathematical model in Equations 1 to 10, Rij in Equation 6 becomes a constant; the chromosome in Equation 7 is a 20-bits binary string. A GA developed in our previous work is applied to find a 20-bits binary string: namely, the one which makes the fitness function or the risk have the smallest value among all possible values of the risk. We randomly choose 20 chromosomes to form a generation. The one with the smallest value of the risk is placed in the beginning of each generation. The probabilities for the mutation and crossover operations are set as 0.1 and 0.6 respectively. A satisfactory solution is derived after 30 generations. The final result is shown in Table 4. The components with the occurrence value 1 have been selected to build the AX and will be shared each other. The components with the occurrence value 0 will be not integrated in the AX for their functionalities have been implemented by the selected components and/or they make a big contribution to the risk. Because all the sub-functionalities have been implemented in the AX, i.e. E in Equation 9 is 0, the fitness function is equal to the risk. The smallest risk achieved is 1.775 with this search result, as shown in Fig. 2.

Table 4. The search result

m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{21}	m_{22}	m_{23}	m_{31}	m_{32}
1	1	1	0	0	1	0	1	0	0
m_{33}	m_{41}	m_{42}	m_{43}	m_{51}	m_{52}	m_{61}	m_{62}	m_{71}	m_{81}
0	0	1	0	0	1	0	0	1	1

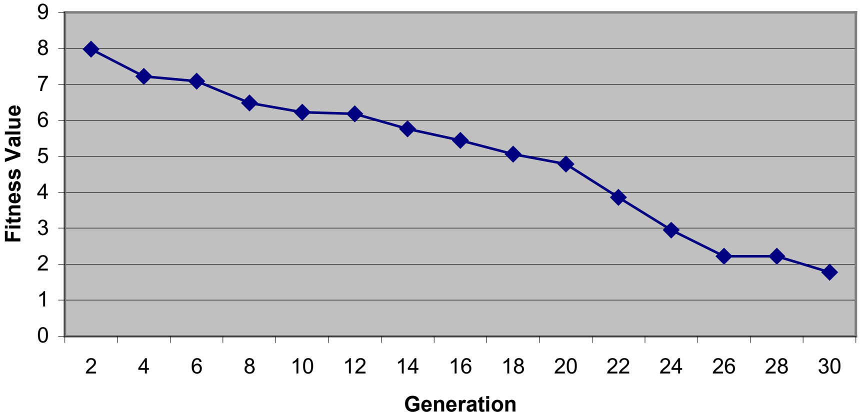


Fig. 2. Fitness values of the GA over the generations

5 Conclusions

This paper discussed the AX based approach to software integration with the special emphasis on the component selection and the risk minimization. We use the concept of the AX for software integration and show that the isolated components can be shared among various participants within the AX. The software component risk is estimated by the frequency of failure multiplied by the severity index. The significant contribution is that the software integration problem has been formulated as a search problem and solved using a GA. A case study from the DFRS is used to illustrate the applicability of the approach. The advantage of using the GA for the software integration is that the method is applicable for any scale problem efficient.

Acknowledgements. The data used in this work was collected from the Derbyshire Fire Rescue Service. Appreciation should be made to their kindly collaboration during the data collection.

Reference

1. Cumming, G., and Hanson, K.: Interaction integration. eAI Journal. April (2002) 28-31.
2. Goldberg, D. E.: Genetic algorithms in search, optimisation and machine learning. Addison-Wesley, Reading, MA (1989)
3. Harman, M., and Jones, B.F.: Search-based software engineering. Information and Software Technology. 43 (2001) 833-839.
4. Jones, B. F., Sthamer, H. H. and Eyres, D. E.: Automatic structural testing using genetic algorithms. Software Engineering Journal. September (1996) 299-306.
5. NASA-STD-8719.13A: Software safety. Nasa Technical Standard. Sept. (1997)
6. Neumann, D.E.: An enhanced neural network technique for software risk analysis. IEEE Transactions on Software Engineering. 28(2002) 904-912.
7. Rao, B.N.: Extreme application integration. eAI Journal. May (2002) 36-40.
8. US MIL_STD_1629A: Procedures for performing failure mode effects and criticality analysis. Nov. (1984)
9. Whitley, D.: A genetic algorithm tutorial, Statistics and computing. 4 (1994) 65-85.
10. Yacoub, S.M., Ammar, H.H.: A methodology for architecture-level reliability risk analysis. IEEE Transactions on Software Engineering. 28(2002) 529-547.
11. Yang, L., Hayes, J., and Gell, M.: Assimilation exchanges and integrated digital environments. Proceedings of the 1st CIRP (UK) seminar on digital enterprise technology. Durham, UK (2002) 303-305.

Iterative Semi-supervised Learning: Helping the User to Find the Right Records

Chris Drummond

Institute for Information Technology,
National Research Council Canada,
Ottawa, Ontario, Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca

Abstract. This paper proposes extending semi-supervised learning by allowing an ongoing interaction between a user and the system. The extension is intended to not only to speed up search for relevant aircraft engine maintenance records but also to help in improving the user's understanding of the problem domain. After the user has identified a small number of relevant records, the system produces a description which generalizes their common properties. If the user is satisfied with the description, the system retrieves more potentially relevant records. The user critiques the items returned, labeling them as relevant or not. The system updates the description using this new labeling information and retrieves more records. The process continues until the user is satisfied that most relevant records have been found. To validate the efficacy of the approach, a set of related maintenance records are collected using the system. These records are compared to those collected without system support.

1 Introduction

This paper proposes to extend semi-supervised learning to allow labeling to be an ongoing process. In the standard approach, information from a large number of unlabelled examples is used to try to overcome poor classification due to a relatively small number of labeled examples. Here, unlabelled examples are also used but instead of a fixed set of labeled examples, the user progressively labels more of them as search continues. Another important difference is that, unlike other semi-supervised learning algorithms [2,6,10], the system generates a description of the common properties of the relevant records. This is advantageous for a number of reasons. Extracting the essential properties of relevant records helps the user to understand the domain and to confirm that the right records have been found. The description is also useful as a basis for finding a different set of related records in the same domain. Further, in this application at least, the description forms the basis of an SQL query which actually retrieves the records from the relational database where they are stored.

Although existing semi-supervised learning algorithms might be easily extended to make them incremental, it would be much harder to modify them to generate the descriptions needed here. Instead, clustering, a traditional unsupervised learning method, is used to generate a tree that represents the degree of similarity between rec-

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1239-1248, 2004.

ords. This tree, in conjunction with labeling information supplied by the user, is used to identify additional relevant records and to guide how rule generalization is applied to these records to generate the description.

The main motivation for this research is the construction of parametric models for fault prediction in turbofan engines on commercial aircraft [9]. Commercial aircraft are costly to maintain. Unscheduled maintenance can increase costs considerably. Failures are often detected while the aircraft is being readied for the next flight. The resultant delays produce customer dissatisfaction and a loss of profit, perhaps to the point where all profit for an individual flight is wiped out. If the airport does not carry the necessary part in stock, a new part may have to be flown in, increasing delays and costs further. Any advanced warning of component failure, which allows repair at a more appropriate juncture, would be of great benefit to an airline.

To support fault prediction, we have access to a large database of maintenance records. But before fault prediction models can be constructed, it is necessary to identify which maintenance records refer to which parts being replaced. Each record consists of a number of related forms which in turn describe the problem first being noticed, the action taken to correct the problem, the part removed and the part that replaced it. What makes the task difficult is that not all the forms are filled out completely on every occasion. Sometimes information is missing; sometimes it is entered into the wrong field. The problem is exacerbated as several different part numbers may be used: the manufacturer□s, the airline□s or possibly numbers from other sources. The numbers are not always entered completely; additional numbers are often included, indicating such things as revision levels. Surprisingly, however, there is no real ambiguity in the part that was actually replaced. There are free text fields in a couple of the forms and the mechanics are careful to detail the action taken.

Once the maintenance record has been located, a user who has sufficient familiarity with the domain will have little doubt in which part was replaced. The difficulty lies in writing a query that returns the appropriate records. Locating relevant records is invariably an iterative process; seldom does an initial query return exactly what the user requires. In the approach proposed here, the user constructs a simple description of at least some of the relevant records. The system converts this into an SQL query and retrieves matching records. The user critiques these records by labeling the relevant ones as positive and the irrelevant ones as negative. The system uses this information to generate a new description. This the user studies, possibly altering it if appropriate. It is then used to generate a new query and retrieve more records. This process repeats until the user is satisfied that all, or at least a significant fraction, of the relevant records have been returned.

This approach to finding relevant records should be useful to support fault prediction modeling in other domains where the down time of equipment is expensive. For instance, in mining and quarrying the heavy trucks often operate 24 hours a day and having one out of service is a serious and costly problem. This approach should also be useful in applications not part of a larger data mining process. There are certainly many situations in which humans collect related records. It might be important to find medical records where patients of similar age have similar symptoms, for example. Another example is configuration management. It might be useful to gather engineering change orders, either for hardware or software, based on the types of problem that lead to particular changes.

2 Supporting a User's Search

This section begins by describing in detail the user interface and how it might be used to search for a set of relevant maintenance records. It then discusses the background processes that support search and how they are initiated as a result of user actions.

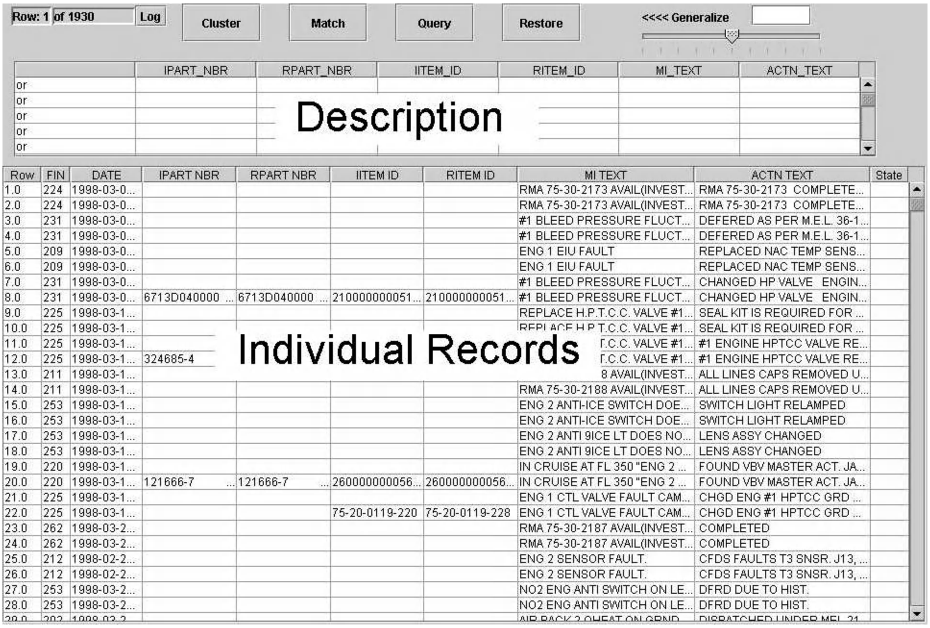
2.1 The User Interface

The user interface, shown in Figure 1, is divided into three main parts. At the top is a series of buttons that in turn control the clustering ("Cluster"), the generation of the description ("Match"), the retrieval of the records ("Query") and the ability to backtrack ("Restore"), at present only to the beginning of search. The upper small table shows the description. The large table occupying the lower three quarters of the interface displays a scrollable list of the individual records that meet the current description. At the beginning of search the description is empty and the list includes all of the maintenance records detailing problems with aircraft engines. For each record, the first three fields indicate the record number, the aircraft on which the problem occurred and the date when it occurred. The next two fields give the manufacturer's part number both for the component that was installed and for the component that was removed. The next two fields give the standard industry number (called the Item Id) for both the installed and removed component. Lastly, there are two free text fields, the first entered by the mechanic who noticed the problem and the second by the mechanic who fixed it.

The description table has equivalent fields for the four part numbers and two text entries. The user can add information directly to these fields and press the "Query" button to retrieve records that match the resultant description. Alternatively, the user can label individual records by pressing the left or right mouse button for positive or negative labels respectively (the middle button removes the label). When the "Match" button is pressed, a description is generated automatically from the labeling information. The user can modify it, if desired Each row of the description table is a single disjunct and pressing the "Query" button returns the union of the records for each row. By clicking the mouse at the beginning of any row, the records that match only that disjunct are retrieved.

Typically, from the author's experience of searching maintenance records using the new system, a user starts with a very simple description, perhaps even a single term. The user studies the matching records to see if they are relevant. If it is clear from the records what common characteristics define relevancy, the user might add extra terms manually to the existing description. If is not clear, the user would label the items that are clearly relevant or irrelevant and review the description generated by the system. Again, the user might directly modify the description or allow the system generated description to expand the search. We would anticipate that, particularly in an unfamiliar domain, initially the user would depend on the system to control the search. However, as familiarity is gained, the user should have more confidence in modifying the description directly.

Search ends when the user is satisfied that a sufficient proportion of the relevant records have been found. What constitutes sufficient is problem dependent. In this application, it is important to have nearly all the relevant records. The records form the basis of training data for additional leaning algorithms to extract fault prediction models. More examples tend to produce more accurate models. A more general concern, likely also true in other domains, is that if a significant number of relevant records are missed there is a danger they may concern problems of a single type and thus an important problem class may be overlooked.



	IPART_NBR	RPART_NBR	IITEM_ID	RITEM_ID	MI_TEXT	ACTN_TEXT
or						
or						
or						
or						
or						

Row	FIN	DATE	IPART NBR	RPART NBR	IITEM ID	RITEM ID	MI TEXT	ACTN TEXT	State
1.0	224	1998-03-0...					RMA 75-30-2173 AVAIL(INVEST...	RMA 75-30-2173 COMPLETE...	
2.0	224	1998-03-0...					RMA 75-30-2173 AVAIL(INVEST...	RMA 75-30-2173 COMPLETE...	
3.0	231	1998-03-0...					#1 BLEED PRESSURE FLUCT...	DEFERED AS PER M.E.L. 36-1...	
4.0	231	1998-03-0...					#1 BLEED PRESSURE FLUCT...	DEFERED AS PER M.E.L. 36-1...	
5.0	209	1998-03-0...					ENG 1 EIU FAULT	REPLACED NAC TEMP SENS...	
6.0	209	1998-03-0...					ENG 1 EIU FAULT	REPLACED NAC TEMP SENS...	
7.0	231	1998-03-0...					#1 BLEED PRESSURE FLUCT...	CHANGED HP VALVE ENGIN...	
8.0	231	1998-03-0...	6713D040000 ...	6713D040000 ...	210000000051...	210000000051...	#1 BLEED PRESSURE FLUCT...	CHANGED HP VALVE ENGIN...	
9.0	225	1998-03-1...					REPLACE H.P.T.C.C. VALVE #1...	SEAL KIT IS REQUIRED FOR ...	
10.0	225	1998-03-1...					REPLACE H.P T.C.C. VALVE #1...	SEAL KIT IS REQUIRED FOR ...	
11.0	225	1998-03-1...					T.C.C. VALVE #1...	#1 ENGINE HPTCC VALVE RE...	
12.0	225	1998-03-1...	324685-4				T.C.C. VALVE #1...	#1 ENGINE HPTCC VALVE RE...	
13.0	211	1998-03-1...					8 AVAIL(INVEST...	ALL LINES CAPS REMOVED U...	
14.0	211	1998-03-1...					RMA 75-30-2188 AVAIL(INVEST...	ALL LINES CAPS REMOVED U...	
15.0	253	1998-03-1...					ENG 2 ANTI-ICE SWITCH DOE...	SWITCH LIGHT RELAMPED	
16.0	253	1998-03-1...					ENG 2 ANTI-ICE SWITCH DOE...	SWITCH LIGHT RELAMPED	
17.0	253	1998-03-1...					ENG 2 ANTI 9ICE LT DOES NO...	LENS ASSY CHANGED	
18.0	253	1998-03-1...					ENG 2 ANTI 9ICE LT DOES NO...	LENS ASSY CHANGED	
19.0	220	1998-03-1...					IN CRUISE AT FL 350 "ENG 2 ...	FOUND VBV MASTER ACT. JA...	
20.0	220	1998-03-1...	121666-7 ...	121666-7 ...	260000000056...	260000000056...	IN CRUISE AT FL 350 "ENG 2 ...	FOUND VBV MASTER ACT. JA...	
21.0	225	1998-03-1...					ENG 1 CTL VALVE FAULT CAM...	CHGD ENG #1 HPTCC GRD ...	
22.0	225	1998-03-1...			75-20-0119-220	75-20-0119-228	ENG 1 CTL VALVE FAULT CAM...	CHGD ENG #1 HPTCC GRD ...	
23.0	262	1998-03-2...					RMA 75-30-2187 AVAIL(INVEST...	COMPLETED	
24.0	262	1998-03-2...					RMA 75-30-2187 AVAIL(INVEST...	COMPLETED	
25.0	212	1998-02-2...					ENG 2 SENSOR FAULT.	CFDS FAULTS T3 SNSR. J13, ...	
26.0	212	1998-02-2...					ENG 2 SENSOR FAULT.	CFDS FAULTS T3 SNSR. J13, ...	
27.0	253	1998-03-2...					NO2 ENG ANTI SWITCH ON LE...	DFRD DUE TO HIST.	
28.0	253	1998-03-2...					NO2 ENG ANTI SWITCH ON LE...	DFRD DUE TO HIST.	
29.0	202	1998-03-2					AIR PACK 2 OHEAT ON GRND	DISPATCHED UNDER MEL 21	

Fig. 1. The User Interface.

2.2 Background Processes

There are three essential background processes: clustering the maintenance records; labeling new potentially relevant records by spreading activation; generating the description shown to the user through rule generalization.

Clustering the Records. A central data structure used by the system is a tree generated by clustering all the maintenance records. The simple agglomerative clustering algorithm □agnes□[7] is used. This takes the two most similar items and combines them into a cluster. The next closest pair, which may be two items or one item and the existing cluster, are combined to form a new cluster. This is repeated combining items and/or clusters until a single cluster remains. This results in a binary tree whose leaves are the individual items. Figure 2 shows the result of clustering a small sample of maintenance records. The numbers at each leaf are the particular record numbers The position on the y-axis of the horizontal lines indicates the degree of similarity when two clusters were merged. At the bottom, close to the leaves, the records and the resultant clusters are similar. As we move up the tree, similarity decreases. In this example there are two main clusters, on the left and right of the dashed line, with smaller internal sub-clusters.

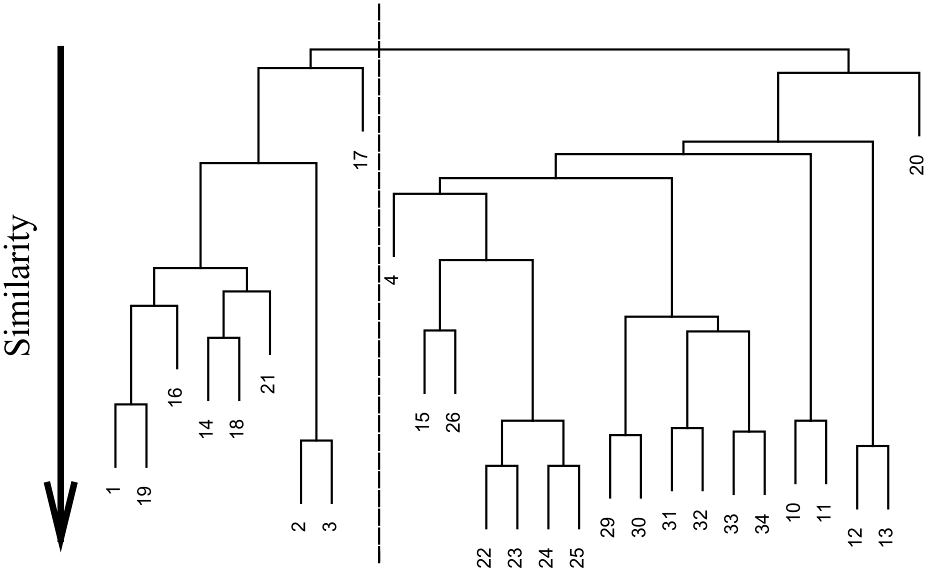


Fig. 2. The Binary Tree.

The agglomerative clustering algorithm requires a similarity measure, which in this application is based on comparing records field-by-field, term-by-term. For each term in one record, we find the best matching term from the same field in the other record. Although an exact match gives the largest score, lower scores are obtained if terms share a common prefix. This is useful not only for part numbers, where leading digits are typically the most important, but also for free text, where words are often abbreviated by leaving off the endings. Text is further processed by removing □stop words□ and any periods occurring in acronyms. Vowels are removed (e.g. VLV = VALVE) as they are often left out in the □short hand□the mechanics use. The total score is then just a count of the number of matching characters. The results are normalized to range from zero to one by using a sigmoid squashing function (2*(1-1/(1+1.1^-x)). The tree produced by the clustering algorithm is loaded into the system prior to any search. Each internal node, representing a merge point, includes the similarity measure; each leaf includes the corresponding record number.

Labeling by Spreading Activation. When the user adds positive and negative labels to records, by clicking the mouse buttons, these are also recorded at the appropriate leaves in the tree. Leaves corresponding to unlabelled records are left unassigned. When the user presses the □Match□button, spreading activation is used through the tree to label both interior nodes and unassigned leaves. The process is analogous to current flow through a resistor network where the resistor values are determined by the similarity measure. A positive label is equivalent to connecting all positively labeled leaves to a positive voltage source and a negative label to connecting all negatively labeled leaves to a negative voltage source. Figure 3 gives a very simple example, where the user has labeled records 1 and 2 as positive and 6 as negative. The current flow through the resistors determines the voltage at any node and therefore the label for the leaves representing the unlabeled records. In this example, if the similarities are roughly equal we would expect record 3 to be labeled positive and records 4 and 5 negative. Interior nodes also have a weak negative bias, indicated by the dotted connections. This allows classification of records when there are only positive labels and controls the level of generalization.

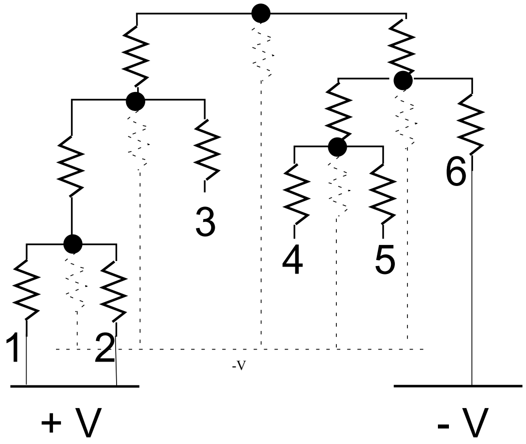


Fig. 3. A Resistor Network.

Generating a Description. Once all nodes in the tree have been labeled, each subtree containing only positive nodes is used to generate a disjunct in the description. Beginning at the leaves of each subtree, pairs of records are generalized by finding common terms. This is done in exactly the same way as the similarity measure was calculated, such as removing vowels and matching prefixes (e.g. RPL matches REPLACED, more examples are given in section 3). The generalized terms (e.g. RPL) now form a conjunctive rule. The rule is further generalized by combining it with rules from other generalized records, or with other new records directly. The process terminates when the root of the subtree is reached or there is no generalization of sufficient complexity possible (a rule must contain at least 10 characters to prevent trivial matches). In the latter case, rules from lower down the subtree are used. The resulting disjuncts are shown to the user on separate lines of the description table. Once the user is happy with the description and presses the □Query□button, the description is converted into an SQL query which retrieves matching records from the database. If the new description does not expand the search sufficiently, the □Generalization□slider at the top of the interface can be used to decrease the negative bias. This tends to increase the effect of positive labels and therefore generalization.

3 Experimental Validation

To experimentally validate the system and to show that some speed up is obtained, a search carried out previously by the author is repeated using the system. The airline had identified a number of components that were felt to be particularly costly in terms of maintenance. Relevant records had been found previously by querying the database directly using SQL queries constructed by hand. The system was built in part with the aim of making this process easier and faster. The rest of this section details the search for records about the replacement of the “low power turbine cooler valve”, typically represented by the acronym LPTC.

On opening the interface, the system displays the nearly 2000 records representing all maintenance records dealing with engine problems. It is somewhat laborious to study so many items and the list is easily reduced by entering a simple description. Two numbers supplied by the airline C25149000, the manufacturer’s part number, and 75-20-0115, the item id, are entered on separate lines in the description table.

C25149000			
	75-20-0115		

The two text fields are left blank. In this discussion, the tables show only numbers for the installed part. Although the numbers can differ for the removed part, they are often the same or missing altogether. So they are not shown for ease of exposition. After pressing the “Query” button, 19 records are retrieved, 3 matching on the manufacturer’s part number and 16 on the item id. So far there is little difference from the process carried out by hand, one minor advantage is that it is not necessary to write an SQL query. Marking all records as positive and pressing the “Match” button produces four disjuncts. The first two correspond largely to the original information entered by the user although they were generated by generalizing actual records.

C25149000	2600000005654		LPT_C
	75-20-0115-00		

One difference is that the first disjunct includes a new Item Id. (2600000005654) along with the manufacturer’s number. This number is actually the airline’s own number rather than the industry standard number we were given. The discovery of such commonly used additional numbers is an important part of understanding the domain. This information did appear in the records returned by hand crafted SQL queries, but the information was buried in the fields of some of the records. The acronym LPT_C also appears, the underscore indicating a possible vowel. The reason for this is made clear if we look at the free text from two of the 19 records.

Maintenance Action 1: REPLACED RH LPTC VLV AS PER AMM FADEC CKC CARRIED OUT

Maintenance Action 2: LPTACC CHANGED

The component typically written as LPTC, as in the first example, is sometimes written as LPTACC, as in the second example. This is generalized by dropping the extra C and allowing a possible vowel as the fourth letter. The second pair of disjuncts are more general than those entered by the user. The first generalizes the item id; the second uses only the free text fields. Matching on all four disjuncts returns 25 records, an additional six potentially relevant ones.

	75-20-011	75-20- ENG R_PL_C	V_LV
		VLV	CH_NG CH_N LPT_CC T_ST

Unfortunately, three of the records refer to the wrong component. This is due to the first disjunct being over-generalized. The Item Id. 75-20-011 is the prefix for both the "high power turbine cooler valve" and the "low power turbine cooler valve". The text fields do not help. The second free text field identifies a valve (V_LV), but both components are valves. The next disjunct is not over-generalized, it specifically mentions changing (CH_NG) the LPTC. When the incorrect records are marked as negative, all others as positive, and the "Match" button pressed, a new set of six disjuncts is returned. Most notably the offending over-generalized disjunct has disappeared. (NB the first two disjuncts continue to appear in the user interface until the end of search but won't be specifically listed from now on).

C24792000	26000000005654	V_LV	LPT
		75-20- LPTC	CH_CK LPT_C
		75-20- ENG LPT	CH_NG ENG LPTC V_LV
		VLV	CH_NG CH_N LPT_CC T_ST

Here the new Item Id. appears again but now accompanied by a different manufacturer's part number (C24792000). Again the system has highlighted variations in the numbering system not mentioned by the airline. Pressing the "Query" button retrieves 30 records all referring to the correct component. The success of the other disjuncts is primarily due to including some variant of the acronym LPTC in the last field, although terms like changed (CH_NG) are important to make sure the component was actually replaced. Marking all records as positive and pressing the "Match" produces 7 disjuncts. The most useful change is the generalization of the first of these disjuncts by relying only on the newly identified numbers and removing all text terms. The extra disjunct (5th row) does little to improve search.

C24792000	26000000005654		
		75-20- LPTC	CH_CK LPT_C
		75-20- ENG LPT	CH_NG ENG LPTC V_LV
		VLV	CH_NG CH_N LPT_CC T_ST
		CH_N R_P_RT V_LV	CH_N S_RV T_ST

By pressing "Query" again, 35 items are retrieved. The system did not generalize sufficiently to find more relevant records after marking these 35 as positive. Using the "Generalization" control tended to over-generalize, matching again on the "high power turbine cooler valve". Generalizing a couple of the existing disjuncts, by manually removing terms from the first text field, retrieved a total of 46 records. Finding more records by removing terms from the problem description may indicate that it was initially wrongly diagnosed and only when a repair was carried out did the component at fault become clear. The total number of records found was one more than had been identified in the original manual search. But two of the records were incorrect, although they both mentioned an LPTC, one that was changed earlier and one that would be changed later. But overall, the system helped the user quickly find relevant records and identified many of the essential terms that were important in the domain.

4 Limitations and Future Work

Having the user critique records returned by a query might be viewed as a form of relevance feedback [3,4] used extensively in information retrieval. Another similarity is that many such systems use spreading activation to find potentially interesting documents [8]. The large body of research into relevance feedback will undoubtedly prove to be a useful source of ideas to improve the retrieval of maintenance records.

Clustering has had somewhat mixed results in information retrieval [5]. So one question warranting further investigation is why it has worked in the experiments discussed here. One answer is that there are significant differences between the application presented here and the general information retrieval task. Firstly, the records consist of multiple fields of different types not only free text Secondly, the text is much more constrained, being limited to a narrow domain, than the text encountered more generally in documents. Thirdly, this paper has strongly stressed the importance of generating a description not normally a concern in information retrieval.

As part the investigation of the effectiveness of clustering, it would be worth looking at how sensitive the system is to the choice of clustering algorithm. This would involve experiments with other commonly used algorithms [7] and perhaps with other more experimental ones [11]. At present, clustering is done only once but it still extracts useful structure that can be used to find relevant records. It might be advantageous, however, if the clustering took into account the labeling information [1]. However, repeating a standard clustering algorithm every time a user labels new records would be computationally very expensive. It would be worth investigating if fast, local modifications to the clusters in response to new labels would be sufficient.

The system found most of the relevant records without direct modification of the description by the user. To find the last 25% of the records, the user experimented with different generalizations of disjuncts already found by the system. Finer □Generalization□control, allowing the user to specify that only particular disjuncts should be generalized, should help this last exploratory stage of the search. So far records have been compared field by field. But quite often information was entered into the wrong field. In addition, particularly with text fields, similar terms might be quite correctly used in different fields. A simple extension would be to look for terms in other fields but reduce the matching score appropriately. This would require a more complex type of description, requiring disjunctions at the term level. But such disjunctions, and perhaps negations, would be useful anyway, producing more compact descriptions which should be more easily understood by the user.

The primary aim of this work is to speed up a user□s search. The description is also meant to be an important aid to understanding the domain. The process has been so far been evaluated by comparing it to prior experience when doing search by hand. More extensive experiments, ideally with many human subjects, would be the ultimate way of verify the goals have been met. The amount of effort needed though is not warranted at present but more experiments by the author with other parts from aircraft engines will be carried out in the near future. In the longer term, experience in working in a completely different domain would give greater confidence in the generality of the approach.

5 Conclusions

This paper has demonstrated a system that helps users find relevant records in a database containing a very large number of irrelevant ones. It also provides a description of what makes a record relevant. This helps the user to understand the domain and to be confident that the records found are indeed relevant.

References

1. S. Al-Harbi and V. Rayward-Smith (2003) The Use of a Supervised k-Means Algorithm on Real-Valued Data with Applications in Health. *Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp 575-581
2. A. Blum and T. Mitchell. (1998) Combining Labeled and Unlabeled Data with Co-training. *Proceedings of the Workshop on Computational Learning Theory*, pp 92-100
3. D. Haines and W. Croft (1993) Relevance Feedback and Inference Networks. *Proceedings of the 16th International Conference on Research and Development in Information Retrieval*, pp 2-11
4. D. Harman (1992) Relevance Feedback Revisited. *Proceedings of the 15th International Conference on Research and Development in Information Retrieval*, pp 1-10
5. M. Hearst and J. Pedersen (1996). Reexamining the Cluster Hypothesis: Scatter/gather on Retrieval Results. *Proceedings of the 19th ACM SIGIR International Conference on Research and Development in Information Retrieval*, pp 76-84
6. T. Joachims (1999) Transductive Inference for Text Classification using Support Vector Machines. *Proceedings of the 16th International Conference on Machine Learning, pp 200—209*
7. L. Kauffman and P. Rousseeuw (1989) *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley
8. G. Salton and C. Buckley (1998) On the use of spreading activation methods in automatic information. *Proceedings of the 11th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 147-160
9. S. L□tourneau, F. Famili and S. Matwin, (1999) Data mining for Prediction of Aircraft Component Replacement. *IEEE Intelligent Systems and their Applications* 14, 6, pp 59-66
10. K. Nigam, A. McCallum and S. Thrun and T. Mitchell (2000) Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*,39, 2/3, pp 103□ 134
11. Z. Su, L. Zhang and Y. Pan (2003) Document Clustering Based on Vector Quantization and Growing-Cell Structure. *Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp 326-336

Semantic Analysis for Data Preparation of Web Usage Mining

Jason J. Jung and Geun-Sik Jo

Intelligent E-Commerce Systems Laboratory,
School of Computer Engineering, Inha University,
253 Yonghyun-dong, Incheon, Korea 402-751
j2jung@intelligent.pe.kr, gsjo@inha.ac.kr

Abstract. As the web usage patterns from clients are getting more complex, simple sessionizations based on time and navigation-oriented heuristics have been restricted to exploit various kinds of rule discovering methods. In this paper, we present semantic analysis approach based on semantic session reconstruction as finding out semantic outliers from web log data. Web directory service is applied to enrich semantics to web logs, categorizing them to all possible hierarchical paths. In order to detect the candidate set of session identifiers, semantic factors like semantic mean, deviation, and distance matrix are established. Eventually, each semantic session is obtained based on nested repetition of top-down partitioning and evaluation process. For experiment, we applied this ontology-oriented heuristics to sessionize the access log files for one week from IRCache. Compared with time-oriented heuristics, more than 48% of sessions were additionally detected by semantic outlier analysis. It means that we can conceptually track the behavior of users tending to easily change their intentions and interests, or simultaneously try to search various kinds of information on the web.

1 Introduction

As the concern for searching relevant information from the web has been exponentially increasing, the very large amount of log data have been generated in web servers. Thus, many applications have been focusing on various ways to analyze them in order to recognize the usage patterns of users and discover other meaningful patterns [1]. For example, on-line newspaper on the web [2], web caching [3], and supporting user web browsing [4], [5] can be told as the domains relevant to analyzing web log data.

In this paper, among the whole steps of web user profiling mentioned in [6], we have taken the session identification for segmenting web log data in consideration. There are mainly two kinds of heuristics for partitioning each user activity into sequences of entries corresponding to each user visit. First, time-oriented heuristics consider temporal boundaries such as a maximum session length or maximum time allowable for each pageview [7]. Second, navigation-oriented heuristics such as HITS, PageRank, and DirectHit take the linkage between pages into account [8].

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1249-1258, 2004.

However, knowledge that is extractable from sessions identified by those heuristics is limited like frequent and sequential patterns represented by URLs. It means that web logs should be sessionized with semantic enrichment based on ontology in order to find out more potential and meaningful information like a user's preference and intention. More importantly, web caching (or proxy) servers have to track streaming URL requests from multiple clients, because they have to increase predictability for prefetching web content that is expected in next request.

Enriching web logs with their corresponding semantic information has been attempted in some studies. Typically, there are two kinds of approaches which are the *information extraction* and *information dimensions* mapping URLs to set of concepts as a feature vector and a specific value, respectively. While both of them apply keywords extracted from an URL's web page to perform semantic enrichment, we present conceptualizing URL information itself by using web directory and introduce representing conceptualized URLs as tree-like information.

2 Data Model of Web Log and Problem Statement

There are several standard data models of web logs such as NCSA, W3C extended, and IRCache format. In this paper we have focused on the IRCache format, which is the NLANR web caching project [11]. Each IRCache log file consists of ten basic fields, for instance, some records of a log file are as follows.

```
1048118411.784    214655    165.246.31.128    TCP_MISS/503    1568    GET
  http://www.intelligent.pe.kr/a.html - NONE/- text/html
1048118421.159    238    218.53.200.251    TCP_MISS/304    261    GET
  http://www.antara.co.id/images/spacer.gif - DIRECT/202.155.27.190 -
```

There are, generally, some problems to analyze these web logs such as their anonymity, rotating IP addresses connections through dynamic assignment of ISPs, missing references due to caching, and inability of servers to distinguish among different visits. Therefore, we note the problem statements concentrated for semantic sessionization in this paper, as follows.

- **Weakness of IP address field as session identifier.** The same IP address field in a web logs (within the time window or not) cannot guarantee that those requests are caused by only one user, and reversely, requests from the different IP addresses can be generated by a particular user. As independent of temporal difference, the sequential or contiguous logs whose IP addresses are equivalent can be more partitioned or more agglomerated.
- **Simultaneous user requests based on multiple intentions.** A user can request more than an URL “at the same time” by using more than a web browser. It means we have to consider multiple intentions of users by classifying mixed logs according to the corresponding semantics.

Each request consists of timestamp, IP address, and URL fields, as shown in Table 1. A URL field is divided into base URL and reminder, which are the host name of web server and the rest part of full URL, respectively. Then, we assume that each URL is semantically characterized by its base URL.

Table 1. Examples

Timestamp	IP Address	URL(BaseURL + Reminder)
$<t_1, t_2, t_3, t_4>$	ip_1	$<b\_url_1+r_1, b\_url_1+r_2, b\_url_1+r_3, b\_url_2+r_4,>$
$<t_5>$	ip_2	$<b\_url_1+r_5>$
$<t_6, t_7, t_8>$	ip_1	$<b\_url_2+r_6, b\_url_1+r_7, b\_url_3+r_7>$

For example, we are given a web log composed of eight requests ordered by timestamps from t_1 to t_8. We denote the URL set of sequential requests by $<\ldots, b\_url_i+r_j, \ldots>$ mapped to the timestamps $<\ldots, t_i, \ldots>$. These logs are partitioned with respect to an IP address ip_i. After partitioning, we compare semantic distance between base URLs in a set of requests, because we regard a semantic session as the sequence of URL having similar semantics. In other words, we investigate if a user's intention is retained or not.

3 Ontology-Oriented Heuristics for Sessionization

In order to semantically recognize what a URL is about, we try to categorize a requested URL based on ontology. Such ontologies are applied with web directories. We therefore can retrieve the tree-like conceptualized URLs, and then measure the similarity between them.

Outlier, generally, is a data object that is grossly different from or inconsistent with the remaining set of data [9]. In order to detect outliers, cluster-based and distance-based algorithms have been introduced [10], [12], [13], [14].

Meanwhile, semantic outlier of streaming web log data in this paper means a particular URL request whose semantic distance with previous sequence logs is over predefined semantic threshold. Thereby, we define these semantic measurements such as semantic mean and semantic distance. Based on semantic outliers detected by these quantified values, semantic sessionization is conducted.

3.1 URL Conceptualization Based on Web Directories

An ontology, a so-called semantic categorizer, is an explicit specification of a conceptualization [15]. It means that ontologies can play a role of enriching semantic or structural information to unlabeled data. Web directories like Yahoo (http://www.yahoo.com) and Cora (http://cora.whizbang.com) can be used to describe the content of a document in a standard and universal way as ontology [16]. Besides, web directory is organized as a topic hierarchical structure that is an efficient way to organize, view, and explore large quantities of information that would, otherwise, be cumbersome [17].

In this paper we assume that all URLs can be categorized by a well-organized web directory service. There are, however, some practical obstacles to do that, because most of web directories are forced to manage a non-generic tree structure in order to avoid a waste of memory space caused by redundant information [18]. For example, some websites for the category Computer Science:Artificial Intelligence:Constraint Satisfaction:Laboratory can also be in the category Education:Universities:Korea:Inha University:Laboratory. We briefly note that problems with categorizing an URL with web directory as an ontology are the following:

- **The multi-attributes of an URL.** An URL can be involved in more than a category. The causal relationships between categories makes their hierarchical structure more complicated. As shown in Fig. 1 (1), an URL can be included in some other categories, named as A or B.

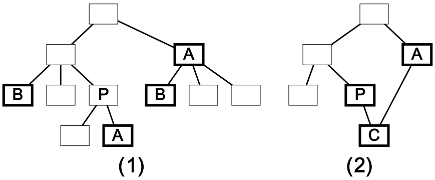


Fig. 1. (1) The multi-attribute of URLs; (2) The subordinate relationship between two categories

- **The relationship between categories.** A category can have more than a path from root node. As shown in Fig. 1 (2), the category C can be a subcategory of more than one like P. Furthermore, some categories can be semantically identical, even if they have different labels.
  - Redundancy between semantically identical categories
  - Subordination between semantically dependent categories

In order to simply handle these problems, we categorize each URL to all possible categories causally related with itself. Therefore, an URL url_i is categorized to a category set $Category(url_i)$, and the size of this category set depend on the web directory. Each element of a category set is represented as a path from the root to the corresponding category on web directory. In Table 1, let the base URLs {b_url$_1$, b_url$_2$, b_url$_3$} semantically enriched to {<a:b:d, a:b:f:k>, <a:c:h>, <a:b:f:j>}. The leftmost concept "a" is indicating the root of web directory and these base URLs are categorized to <d, k>, <h>, and <j>, respectively. In particular, due to multi-attribute of base URL b_url$_1$, *Category*(b_url$_1$) is composed of two different concepts.

3.2 Preliminary Notations and Definitions

We define semantic factors measuring the relationship between two log data. All possible categorical and ordered paths for the requested URL, above all, are obtained, after conceptualizing this URL by web directory. Firstly, the semantic distance is formulated for measuring the semantic difference between two URLs. Let an URL url_i categorized to the sets $\{path_i \mid path^m{}_i \in Category(url_i),\ m \in [1, \ldots, M]\}$ where M is the number of total categorical paths. As simply extending Levenshtein edit distance [19], the semantic distance $\Delta^\Diamond$ between two URLs url_i and url_j is given by

$$\Delta^\Diamond[url_i, url_j] = \arg \min_{m=1,n=1}^{M,N} \frac{\min\left((L_i^m - L_C^{(m,n)}), (L_j^n - L_C^{(m,n)})\right)}{\exp\left(L_C^{(m,n)}\right)} \tag{1}$$

where $L^m{}_i$, $L^n{}_j$, and $L^{(m,n)}{}_C$ are the lengths of $path^m{}_i$, $path^n{}_j$, and common part of both of them, respectively. As marking paths representing conceptualized URLs on trees, we can easily get this common part overlapping each other. The $\Delta^\Diamond$ compares all combination of two sets ($|path_i| \times |path_j|$) and returns the minimum among values in the interval [0, 1], where 0 stands for complete matching. Exponent function in denominator is used in order to increase the effect of $L^{(m,n)}{}_C$. Second factor is to aggregate URLs during a time interval. Thereby, semantic distance matrix $D_{\Delta^\Diamond}$ is given by

$$D_{\Delta^\Diamond}(i,j) = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \Delta^\Diamond[url_{t_i}, url_{t_j}] & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{2}$$

where the predefined time interval T is the size of matrix and diagonal elements are all zero. Based on $D_{\Delta^\Diamond}$, the semantic mean $\mu^{\text{♫}}$ is given by

$$\mu^\Diamond(t_1, \ldots, t_T) = \frac{2\sum_{i=1}^{T}\sum_{j=1}^{T} D_{\Delta^\Diamond}(i,j)}{T(T-1)} \tag{3}$$

where $D_{\Delta^\Diamond}(i, j)$ is the (i, j)-th element of distance matrix. This is the mean value of upper triangular elements except diagonals. Then, with respect to the given time interval T, the semantic deviation $\sigma^{\text{♫}}$ is derived as shown by

$$\sigma^\Diamond(t_1, \ldots, t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=1}^{T}\left(D_{\Delta^\Diamond}(i,j) - \mu^\Diamond(t_1, \ldots, t_T)\right)^2}{T(T-1)}} \tag{4}$$

These factors are exploited to quantify the semantic distance between two random logs and statistically discriminate semantic outliers such as the most distinct or the N distinct data from the rest in the range of over pre-fixed threshold, with respect to given time interval.

3.3 Semantic Outlier Analysis for Sessionization

When we try to segment web log dataset, log entries are generally time-varying, more properly, streaming. In case of streaming dataset, not only semantic factors in a given interval but also the distribution of the semantic mean $\mu^\Diamond$ is needed for sessionization. This will be described in the Sect. 4. We, hence, simply assume that a given dataset is time-invariant and its size is fixed in this section.

In order to analyze semantic outlier for sessionization, we regard the minimization the sum of partial semantic deviation $\mu^\Diamond$ for each session as the most optimal partitioning of given dataset. Thereby, the principle session identifiers $PSI = \{psi_a \mid a \in [1,\ldots, S\text{-}1], psi_a \in [1,\ldots, T\text{-}1]\}$ is defined as the set of boundary positions, where the variables S and T are the required number of sessions and the time interval, respectively.

The semantic outlier analysis for sessionizing static logs SOA_S as objective function with respect to PSI is given by

$$SOA_S(PSI) = \sum_{i=1}^{S} \mu_i^\Diamond \tag{5}$$

where $\mu_i^\Diamond$ means partial semantic deviation of i^{th} segment. In order to minimize this objective function, we scan the most distinct pairs, in other words, the largest value in the semantic distance matrix $D_{\Delta^\Diamond}$, as follows:

$$\mu_{MAX}^\Diamond [T_a, T_b] = \arg \max_{i=1,j-1}^{T} D_{\Delta^\Diamond}(i, j) \tag{6}$$

where arg $\max^{T}_{i=1}$ is the function returning the maximum values during a given time interval $[T_a, T_b]$. When we obtain $D_{\Delta^\Diamond}(p, q)$ as the maximum semantic distance, we assume there must be at least a principle session identifier between p^{th} and q^{th} URLs. Then, the initial time interval $[T_a, T_b]$ is replaced by $[T_p, T_q]$, and the maximum semantic distance in reduced time interval is scanned, recursively. Finally, when two adjacent elements are acquired, we evaluate this candidate psi by using $SOA_S(psi)$. If this value is less than $\sigma^\Diamond$, this candidate psi is inserted in PSI. Otherwise, this partition by this candidate psi is cancelled. This sessionization process is top-down approaching, until the required number of sessions S is found. Furthermore, we can also be notified the oversessionization, which is a failure caused by overfitting sessionization, detected by the evaluation process $SOA_S(PSI)$.

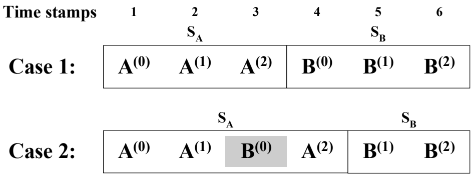


Fig. 2. Top-down approaching sessionization. The four largest semantic distances $\Delta^\Diamond[A^{(1)}, B^{(1)}]$, $\Delta^\Diamond[A^{(2)}, B^{(2)}]$, $\Delta^\Diamond[A^{(2)}, B^{(0)}]$, and $\Delta^\Diamond[A^{(2)}, B^{(1)}]$ are 0.86, 0.85, 0.81, and 0.79, respectively. We want to segment them into two sessions.

As an example, let a URL entry composed of two sessions S_A and S_B in two cases, as shown in Fig. 2. We assume that the semantic distances between $A^{(i)}$s (or $B^{(i)}$s) is much less than between each other. In the first case (Case 1), due to the maximum distance $\Delta^\Diamond[A^{(1)}, B^{(1)}]$ in the initial time interval [1, 6], time interval is reduced to [2, 5], and then, $\Delta^\Diamond[A^{(2)}, B^{(0)}]$ in updated time interval determines that psi_3 can be a candidate.

Finally, the evaluation $\sigma^\Diamond[1, 3] + \sigma^\Diamond[4, 6] \geq \sigma^\Diamond[1, 6]$ makes a candidate psi_3 inserted to *PSI*. This case is clear to find the candidate *psi* and prove this sessionization to be validated. More complicatedly, in second case (Case 2), a heterogeneous request $B^{(0)}$ is located in the session S_A. The first candidate psi_3 is generated by $\Delta^\Diamond[B^{(0)}, A^{(2)}]$ in time interval firstly refined by $\Delta^\Diamond[A^{(1)}, B^{(1)}]$. By the evaluation $\sigma^\Diamond[1, 3] + \sigma^\Diamond[4, 6] < \sigma^\Diamond[1, 6]$, however, this candidate psi_3 is removed. Finally, because the second candidate psi_4 by $\Delta^\Diamond[A^{(2)}, B^{(1)}]$ meets the evaluation $\sigma^\Diamond[1, 4] + \sigma^\Diamond[5, 6] < \sigma^\Diamond[1, 6]$, a candidate psi_4 can be into *PSI*. According to the required number of sessions, this recursion process can be executed.

4 Session Identification from Streaming Web Logs

Actually, on-line web logs are continuously changing. It is impossible to consider not only the existing whole data but also streaming data. We define the time window W as the pre-determined size of considerable entry from the most recent one. Every time new URL is requested, this time window has to be shifted. In order to semantic outlier analysis of streaming logs, we focus on not only basic semantic factors but also the distribution of the semantic mean with respect to time window, $\mu^\Diamond(W^{(T)})$.

As extending SOA_S, the objective function for analyzing semantic outlier of dynamic logs SOA_D is given by

$$SOA_D^{W^{(i)}}(PSI) = \sum_{i=1}^{S} \mu_k^\Diamond \,|_{W^{(i)}} \tag{7}$$

where the $W^{(i)}$ means that the time window from i^{th} URL is applied. We want to minimize this $SOA_D(PSI)$ by finding the most proper set of principle session identifiers. The candidate psi_i is estimated by the difference between the semantic means of contiguous time windows and predefined threshold ε, as shown by

$$\left|\mu_k^\Diamond(W^{(i)}) - \mu_k^\Diamond(W^{(i-\tau)})\right| \geq \varepsilon \tag{8}$$

where τ is the distance between both time windows and assumed to be less than the size of time window $|W|$. Similar to the evaluation process of SOA_S, once a candidate psi_i is obtained, we evaluate it by comparing $SOA^{W(i)}{}_D$ and $SOA^{W(i-1)}{}_D$. Finally, we can retrieve *PSI* to sessionize streaming web logs. In case of streaming logs, more particularly, a candidate *psi* meeting the evaluation process can be appended into unlimited size of *PSI*.

5 Experiments and Discussion

For experiments, we collected the sanitized access logs from sv.us.ircache.net, one of web cache servers of IRCache. These files were generated for seven days from 20 March 2003 to 26 March 2003. Raw data consist of 11 attributes and about 9193000 entries. The total size of them is about 1.2 GB. We verified sessionizing process proposed in this paper on a PC with a 1.2 GHz CPU clock rate, 256 MB main memory, and running FreeBSD 5.0. During data cleansing, logs whose URL field is ambiguous (wrong spelling or IP address) are removed, as referring to web directory.

We compared two sessionizations based on time oriented and ontology oriented heuristics, with respect to the number of segmented sessions and the reasonability of association rules extracted from them. In case of ontology-oriented sessionization, fields related with time such as "Timestamp" and "Elapsed Time" were filtered. Time-oriented heuristics simply sessionized log entries between two sequential requests whose difference of field "Timestamp" is more than 20 milliseconds with respect to the same IP address. On the other hand, for ontology-oriented heuristics, the size of time window *W* was predefined as 50.

Table 2. The number of sessions by time-oriented heuristics and ontology-oriented heuristics (static and dynamic logs) from logs for seven days (20-36 March 2003).

	1	2	3	4	5	6	7
Time-oriented	1563	1359	1116	877	1467	1424	1384
Ontology-oriented (Static logs, SOA_S)	907 (58%)	923 (68%)	692 (62%)	421 (48%)	807 (55%)	783 (55%)	844 (61%)
Ontology-oriented (Dynamic logs, SOA_D)	983 (63%)	1051 (77%)	939 (84%)	683 (78%)	1118 (76%)	827 (58%)	1105 (80%)
Common Session Boundary	47%	51%	49%	48%	57%	32%	74%

The numbers of sessions generated in both cases are shown in Table 2. Time-oriented heuristics estimate denser sessionization than two ontology-oriented approaches. It means that ontology-oriented heuristics based on SOA_S or SOA_D, generally, can make URLs requested over time gap semantically connected each other. They, SOA_S or SOA_D, decreased the number of sessions to, overall, 58.14% and 73.71%, respectively, compared to time-oriented heuristics. Even though ontology-oriented heuristics searched fewer sessions, the rate of common session boundaries (the number of common sessions matched with time-oriented heuristics over the number of sessions of SOA_D) is average 51.1%. It shows that more than 48% of sessions not segmented by time-oriented heuristics can be detected by semantic outlier analysis. While time oriented sessionization is impossible to recognize patterns of users who is easily changing their preferences or simultaneously trying to search various kinds of information on the web, ontology-oriented method can discriminate these complicated patterns.

Table 3. Evaluation of the reasonability of the extracted ruleset (hit ratio (%))

	1	2	3	4	5	6	7
Time-oriented	0.06	0.32	0.46	0.41	0.51	**0.52**	0.49
Ontology-oriented (Static logs, SOA_S)	0.05	0.45	0.66	0.72	**0.76**	0.74	0.75
Ontology-oriented (Dynamic logs, SOA_D)	0.05	0.46	0.52	0.67	0.70	**0.75**	0.72

We also evaluated the reasonability of the rules extracted from three kinds of session sequences. According to the standard *least recently used* (LRU), we organized the expected set of URLs, which means the set of objects that cache server has to prefetch. The size of this set is constantly 100. As shown in Table 3, we measured the two hit ratios by both of their sessionizations for seven days. The maximum hit ratios in three sequences were obtained 0.52, 0.76, and 0.75, respectively. Ontology-oriented sessionization SOA_S acquired about 24.5% improvement of prefetching performance, compared with time-oriented. Moreover, we want to note that the difference between SOA_S and SOA_D. For the first three days, the hit ratio of SOA_S was higher than that of SOA_D by over 5%. Because of streaming data, SOA_D showed the difficulty in initializing the ruleset. After initialization step, however, the performances of SOA_S and SOA_D were converged into a same level.

6 Conclusions and Future Work

In order to mine useful and significant association rules from web logs, many kinds of well-known association discovering methods have been developed. Due to the domain specific properties of web logs, sessionization process of log entries is the most important in a whole step. We have proposed ontology-oriented heuristics for sessionizing web logs. In order to provide each requested URL with the corresponding semantics, web directory service as ontology have been applied to categorize this URL. Especially, we mentioned three practical problems for using real non-generic tree structured web directories like Yahoo. After conceptualizing URLs, we measured the semantic distance matrix indicating the relationships between URLs within the predefined time interval. Additionally, factors like semantic mean and semantic deviation were formulated for easier computation.

We considered two kinds of web logs which are stationary and streaming. Therefore, two semantic outlier analysis approaches SOA_S and SOA_D were introduced based on semantic factors. Through the evaluation process, the detected candidate semantic outliers were tested whether their sessionization is reasonable or not. According to results of our experiments, investigating semantic relationships between web logs is very important to sessionize them. Classifying semantic sessions, 48\% of total sessions, brought about 25% higher prefetching performance, compared with time-oriented sessionization. Complex web usage patterns seemed to be meaninglessly mixed along with “time” could be analyzed by ontology.

In future work, we have to consider bottom-up approaching sessionization to cluster sessions divided by top-down approaching. As exploiting semantic sessionization proposed in this paper to the web proxy server, more practically, we will study association rule mining on various web caching architecture, in order to improve the predictability of content prefetching.

Acknowledgement. This work was supported by the Korea Science and Engineering Foundation(KOSEF) through the Northeast Asia e-Logistics Research Center at University of Incheon.

References

1. Cooley, R., Srivastava, J., Mobasher, B.: Web Mining: Information and Pattern Discovery on the World Wide Web. In Proc. of the 9th IEEE Int. Conf. on Tools with Artificial Intelligence (1997)
2. Batista, P., Silva, M.J.: Web Access Mining from an On-line Newspaper Logs. In Proc. 12th Int. Meeting of the Euro Working Group on Decision Support Systems (2001)
3. Bonchi, F., Giannotti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D., Renso, C., Ruggieri, S.: Web log data warehousing and mining for intelligent web caching. Data and Knowledge Engineering **39**(2) (2001) 165–189
4. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems **1**(1) (1999) 5–32
5. Berendt, B., Spiliopoulou, M.: Analysing navigation behaviour in web sites integrating multiple information systems. The VLDB Journal **9**(1) (2000) 56–75
6. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Communications of the ACM **43**(8) (2000)
7. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. In Proc. of the 4th WebKDD Workshop at the ACM-SIGKDD Conf. on Knowledge Discovery in Databases (2002)
8. Chen, Z., Tao, L., Wang, J., Wenyin, L., Ma, W.-Y.: A Unified Framework for Web Link Analysis. In Proc. of the 3rd Int. Conf. on Web Information Systems Engineering (2002) 63–72
9. Han, J., Kamber, M.: Data Mining: *Concepts and Techniques*. Morgan Kaufmann Publishers (2001)
10. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets. In Proc. of the ACM SIGMOD Conf. on Management of Data (2000) 427–438
11. IRCache Users Guide. http://www.ircache.net/
12. Arning, A., Agrawal, R., Raghavan, P.: A Linear Model for Deviation Detection in Large Databases. In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, (1996) 164–169
13. Menasalvas, E., Millan, S., Pena, J.M., Hadjimichael, M., Marban, O.: Subsessions: a granular approach to click path analysis. In Proc. of the IEEE Int. Conf. on Fuzzy Systems (2002) 878–883
14. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining Access Patterns Efficiently from Web Logs. In Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00) (2000)
15. Gruber, T.: What is an Ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
16. Labrou, Y., Finin, T.: Yahoo! as an Ontology: Using Yahoo! Categories to Describe Documents. In Proc. of the 8th Int. Conf. on Information Knowledge Management (1999) 180–187
17. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Building Domain-Specific Search Engines with Machine Learning Techniques. AAAI Spring Symposium (1999)
18. Jung, J.J., Yoon, J.-S., Jo, G.-S.: Collaborative Information Filtering by Using Categorized Bookmarks on the Web. In Proc. of the 14th Int. Conf. on Applications of Prolog (2001) 343–357
19. Levenshtein, I.V.: Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, **10**(8) (1966) 707–710
20. Aggarwal, C., Wolf, J.L., Yu, P.S.: Caching on the World Wide Web. IEEE Tran. on Knowldge and Data Engineering **11**(1) (1999) 94–107

Prediction of Preferences through Optimizing Users and Reducing Dimension in Collaborative Filtering System

Su-Jeong Ko

Mendus Information System Co.
Bangbae-dong, Sucho-gu, Seoul, Korea..
{sujeongko@yahoo.co.kr}

Abstract. Collaborative filtering systems employ statistical techniques to find a set of users known as neighbors, who have a history of agreeing with the target user. However, the problems associated with high dimensionality in the recommender systems have been discussed in several studies. In addition, the degree of correlation is computed between only two users. Although the preference correlation of two users may not be very high, their preferences can serve as useful data for preference prediction. The preference information of the two users, however, cannot be used to give a recommendation because the degree of their mutual correlation is low. The users preferences in collaborative filtering systems are not necessarily accurate information. Carelessly entered information stored in the collaborative filtering database must be excluded. In this paper, Entropy is used for optimizing users. Bayesian classification shall be used to classify items to lower the dimension of the user-item matrix in this paper. To extract the features of items this paper uses association word mining. Since this method is representing document as a set of association words, it prevents users from being confused by word sense disambiguation. Also, the users can be put into clusters by using the genetic algorithm on the classified items to solve the problems of previous clustering algorithm such as the need to predefine the number of clusters, the high sensitivity to noise in data, and the possibility their resulting data would converge with the region□s optimal solution. Finally, it predicts typical preferences by using entropy to solve the problem depending on the correlation match between only two users. The typical preferences mean pseudo preferences that represent preferences of items within the group. A dynamic recommendation for an item is made using the typical preference. To give dynamic recommendations to users, the process of classifying users receiving recommendations into the most suitable group takes precedence.

1 Introduction

Collaborative filtering recommender systems, e.g. Ringo[8,24], Tapestry system [6], and GroupLens research system[9,19], have been successful. However, their common use has exposed some of their limitations such as the problem of sparsity in the data set. The sparsity problem is associated with high dimensionality. The sparsity problem in recommnder system has been addressed in [7, 21]. The problem

R. Orchard et al. (Eds.): IEA/AIE 2004, LNAI 3029, pp. 1259-1268, 2004.

with high dimensionality in recommender systems have been discussed in [4], and application of dimensionality reduction techniques to address these issues has been investigated in several studies [20]. Secondly, the degree of correlation is computed between only two users. Although the preference correlation of two users may not be very high, their preferences can serve as useful data for preference prediction. The preference information of the two users, however, cannot be used to give a recommendation because the degree of their mutual correlation is low [20]. One such study explores a method that uses one of the EM algorithms [16], K-means algorithm [2, 11], entropy weighting, and SVD. These methods group users by the feature selection of a group [4, 7, 25]. The research proposing a cluster feature selection method, called ENTROPY, finds important attributes for each cluster based on entropy weighting and SVD. However, it is hard to evaluate the worth of original attributes because of information loss of a converted data set by SVD. There have been researches to overcome these weak points of collaborative filtering such as the K-Nearest Neighbor method and clustering. The K-means clustering method is one of the clustering techniques and performs well on numeric data sets in general [6, 16, 18]. However, we assert that in order to predict user preferences more accurately, we should consider both high and low similarities of each user, since a user who has contradicting similarities may give valuable information in prediction. It cannot thus address the problem of sparsity[15]. The method has another shortcoming, though, in that recommendations are made depending on the correlation match between only two users, and it cannot be made when there is a low degree of preference correlation [18]. User preferences are not necessarily accurate information. Carelessly entered information stored in the collaborative filtering database must be excluded. Entropy is used for this purpose.

The proposed method in this paper uses the entropy to optimize users. After calculating entropies of users in collaborative filtering database, users with the value less than a threshold are excluded from the database. Bayesian classification shall be used to classify items to lower the dimension of the user-item matrix. Also, by using the genetic algorithm on the classified items, the users can be put into clusters. Previous clustering algorithms showed problems such as the need to predefine the number of clusters, the high sensitivity to noise in data, and the possibility their resulting data would converge with the region□s optimal solution [18, 19]. In this paper, to rid of these problems, clustering is done with the genetic algorithm. The proposed method in this paper extracts features from items. The native feature space consists of unique words with single dimension when it occurs in documents, which can be tens or hundreds of thousands of words for even a moderate-sized text collection. This is prohibitively high for many learning algorithms. Since the feature extraction method using association word mining automatically generates noun phrases by using the Apriori algorithm without calculating the probability for index. After optimizing users and reducing dimensions, the proposed method predicts typical preferences by using entropy to solve the problem depending on the correlation match between only two users. The typical preferences mean values that represent preferences of items within the group. This method makes dynamic recommendation possible and improves the accuracy of recommendation by comparing active user with not all users in a group but a pseudo user with typical preferences. The proposed method is tested on a database of user evaluated web documents, and the test result demonstrates that the proposed method is more effective than previous methods as a matter of recommendation.

2 Optimizing Collaborative Users

The user-item matrix used in collaborative filtering recommendations will be expressed as the matrix $R=\{r_{ij}\}(i=1,2,\ldots,n \text{ and } j=1,2,\ldots,m)$. The Matrix R is composed of a set with an m number of □items□ a set with an n number of users, and r_{ij}, the value of the user u_i □s preference for the item d_j. Specifically, the set of items is expressed as $I=\{d_j\}(j=1,2,\ldots,m)$, and the set of users is expressed as $U=\{u_i\}(i=1,2,\ldots,n)$. To learn the preferences of the user, it is necessary to classify the items he or she has rated and has shown interest in. Interest levels are represented on a scale of 0~1.0 in increments of 0.2, a total of 6 degrees. Only when the values are higher than 0.5 the users are classified as showing interest. The items are web-documents, which are related to computers and gleaned by an http down-loader.

The distribution of user preferences for items is very significant in the collaborative filtering system. For instance, if a user has rated all items 0.8, it is impossible to predict the typical preferences based on this value. If all items are rated 0.8, it is very probable that the user rated the items very carelessly because of the tedious rating system or lack of time. Thus, since these data cannot accurately represent the user□s preferences, using these data to predict the typical preferences will decrease the accuracy of recommendations. The typical preferences of users must be based on the more carefully chosen preferences of the users. Carefully chosen user preferences are more accurate than identical preferences because users cannot have identical tastes for all items in which they are interested. If the values of the preferences of users within a group range uniformly between 0 and 1, such data can be used to predict the typical preferences within the group. If the values are identical, however, such data will decrease the accuracy of the recommendation. To apply the abovementioned theory[14], entropy is used to predict the typical preferences of users within a group. The process of optimizing collaborative users using entropy is composed of 3 steps. In Step 1 the absolute preferences are converted into the relative preferences. In Step 2 the entropies of the collaborative users are computed. In Step 3 collaborative users are optimized. In Step 1 when the entropies of collaborative users are computed based on the absolute preferences, accurate values cannot be computed because the computation is greatly influenced by the absolute values of the preferences rather than the preferences distribution. Thus, the process of converting the absolute preferences into the relative preferences is required.

In Step 2, the entropies of collaborative users are obtained based on the relative preferences converted. Equation (1) computes the entropy (H_{cui}) of a user within a group. $Rp_{cui,j}$ in Equation (1) indicates the relative preferences of the collaborative user cu_i for the item j.

$$H_{cui} = -\sum_{j} Rp_{cui,j} \cdot log_{2} Rp_{cui,j} \tag{1}$$

In Step 3 collaborative users are optimized based on high numerical values. In this step, the threshold for the entropy of the collaborative users is set, users beyond the threshold are extracted, and users with low entropy are excluded. This paper has set the threshold at 1 and excluded collaborative users with entropies under 1 in the typical preferences prediction.

3 Prediction of Preference through Reducing Dimension of Collaborative Matrix

In collaborative filtering, both the Naïve Bayes classifier and the genetic algorithm [17] are used to decrease the dimension of the matrix. Previous clustering algorithms have shown the problems; e.g. the need to predefine the number of clusters, the high sensitivity to noise in data, and the possibility that their resulting data would converge with the region's optimal solution [12,23]. The proposed method in this paper, to rid of these problems, the clustering is done with a genetic algorithm. The combination of the Naïve Bayes with the genetic algorithm solves the problem of sparsity without reducing the accuracy of recommendation because the Naïve Bayes classifier classifies items based on their semantic relations and then the genetic algorithm clusters users into groups based on the classified items with the reduced dimension. Therefore, the combination reduces the amount of the computing time taken by the genetic algorithm. To extract the features of documents, association word mining method is used in this paper. Since this method is representing document as a set of association words, it has advantages of word sense disambiguation and representing documents in detail.

In this section, the proposed method will present a method of clustering users using the genetic algorithm. The matrix defined as R, must first be translated to R_1. The matrix $R_1=\{c_{ki'}\}(k=1,2...N \text{ and } i'=1,2...n)$ is a {class-user} matrix, where R_1, $C=\{class_k\}(k=1,2,...,N)$ is defined as a set of items divided into classes. The value c_{ki} of matrix R_1 is the number of items classified as $class_k$, which the user $u_{i'}$ has rated and shown interest in. Using these definitions, Equation (2) must also be defined in order to arrive at the values of matrix R_1. When the value r_{ij} of the item $d_j(0<j<=m)$ in matrix R is greater than 0.5 and it belongs to the class $class_k$, the value of $c_{ki'}$ in matrix R_1 is increased accordingly by increments of one.

$$R_1=\{ d_j \in class_k, 0< j \le m \,\middle|\, c_{ki'} = c_{ki'} + 1, \forall r_{ij} > 0.5 \} \tag{2}$$

With the structure of matrix R_1 serving as the model, the genetic algorithm clusters users into groups. The genetic algorithm uses genes, chromosomes, and populations to cluster users. The populace evolves by means of initialization, fitness calculation, recomposition, selection, crossover, mutation, and evaluation.

Initialization is the phase where each entity is expressed. The genetic algorithm does not straightforwardly search for a solution to the given problems. It tries to solve the problems by encoding multiple solutions into chromosomes, and by subjecting the solutions to evolutionary operators who choose the most appropriate solution. Because the solutions to the problem undergo an encoding process into chromosomes, it must be recognized that the encoding process of entities determines the space for the solutions. The applicable evolutionary operators as well as the search field created by the evolutionary operators. For these reasons, entity expression is critical to efficient searches. Chromosomes in genetic algorithms are expressed as binary strings, real vectors, and tree structures. To best express the matrix, it will be expressed in real numbers. The fitness of chromosome is calculated to derive the optimal solution.

In genetic algorithms, each entity is assigned a fitness value by which to compare the performance of the entity with that of others. In this paper, the fitness value shall be calculated to aid the search of classes with a similar distribution to that of the target item class, and also to optimize user groups. To cluster users, it is necessary to find users with high values of similarity. User similarity is calculated by using the Jaccard method[19] presented. Originally the Jaccard method is used in information retrieval to find the similarity of documents.

In the recomposition phase, the value of the object function used to calculate the fitness value must be recomposed with a different value. The objective of recomposition is to apply fitness as the probability that affects the selection operator. In the proposed method, Equation (3) is used to control the fitness. Equation (3) calculates the ratio between the total fitness of all chromosomes and the fitness of single chromosome. In Equation (3) $Fitness(c'_{ki'})$ is the value of single chromosomes and $Fitness\_s(c'_{ki'})$ the value of the recomposed chromosomes.

$$Fitness\_s(c'_{ki'}) = \frac{Fitness(c'_{ki'})}{\sum_{class=1}^{N} Fitness(c'_{ki'})} \tag{3}$$

In the selection phase, the selection operator uses the recomposed fitness values as the basis to choose chromosomes for crossover. In the crossover phase, the crossover operators use crossover rates as the basis to choose chromosomes for crossover. Crossover is done by randomly choosing a single splicing point on a gene, then, by replacing the gene segments with the gene in the corresponding place of another gene in a 1-point crossover method. The typical range of crossover rates is between 0.7 and 0.9, and in this proposed method the crossover value of 0.9 is used. In the mutation phase, the value of a single bit is changed according to the designated probability. The mutation rate is defined as the probability that the gene of a chromosome will mutate to a different value. Mutation rates are typically set between 0.01 and 0.05. In this experiment, the mutation rate is set to 0.01. In the evaluation phase, the decision is made as to whether or not to continue evolution. Evolution terminates if the average fitness is equal to or greater than the threshold value of fitness; if not, evolution continues. In this experiment, the threshold value of fitness is set to 1, and if the fitness value is less than 1, evolution continues; if equal to or greater than 1, evolution terminates. Fig. 1 shows the change of average fitness in the first through hundredth generation. Fig. 2 shows the time required for evolution from the first through hundredth generations. In this proposed method the genetic algorithm was run on a Sambo SMP 6400 Qp server, running Windows NT with a 2-way CPU and 256MB of main memory. On the aforementioned system, it took 9.13 seconds to optimally cluster the first through hundredth generations.

Chromosomes at the hundredth generation that have gene values equal to or greater than 1, are chosen for a cluster, and those with a value of 0 are discarded. With users of the first cluster excluded, the rest of the users repeat the process of initialization, fitness calculation, recomposition, selection, crossover, mutation, and evaluation. By repeating this process, it is possible to classify the users into several clusters.

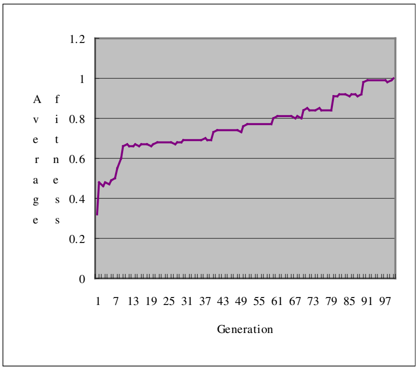


Fig. 1. The change of average fitness

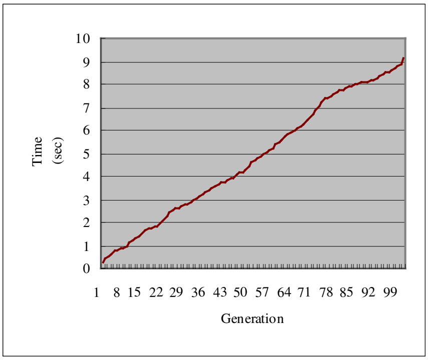


Fig. 2. The time required for evolution from the first through hundredth generations

A dynamic recommendation for items is made using the typical preference. To give dynamic recommendations to users, the process of classifying users receiving recommendations into the most suitable group takes precedence. The preferences are predicted by multiplying the users□absolute preferences with their entropy weighting. Thus, this paper uses entropy weighting(wH_{cui}). wH_{cui} is a user□s entropy weighting that means a probability of each item entropy for entropy sum of all items. To predict preferences of items, the entropy weight of a collaborative user must be merged with the user□s absolute preference. The missing values in collaborative filtering matrix are estimated through absolute preferences being reflected the entropy values. The equation for this is shown in Equation (4), which is the typical preference (Rd_j) of the item d_j.

$$Rd_j = \sum_i p_{cui,j} \cdot wH_{cui} \tag{4}$$

Equation (4) sums up all the items of the absolute preferences of all collaborative users within the group for item d_j to be multiplied with the entropy weighting of the collaborative users.

4 Performance Evaluation

The database for collaborative filter recommendations was created from the data of 200 users and 1600 web documents. Users evaluated a minimum of 10 of the 1600 web documents. The database for content_based filter recommendations was created from 1600 web documents. These 1600 web documents were collected from computer related URLs by an http downloader, then hand-classified into 8 areas of computer information. The 8 areas were classified under the labels of the following classes: {Games, Graphics, News and media, Semiconductors, Security, Internet, Electronic publishing, and Hardware}. The basis for this classification comes from search engines such as AltaVista and Yahoo that have statistically analyzed and classified computer related web documents. Of the 200 users, 100 were used as the training group, and the remaining users were used as the test group. In this paper, mean absolute error (MAE) and rank score measure(RSM), both suggested by paper [5] are used to gauge performance. MAE is used to evaluate single item recommendation systems. RSM is used to evaluate the performance of systems that recommend items from ranked lists. The accuracy of the MAE, expressed as Equation (5), is determined by the absolute value of the difference between the predicted value and real value of user evaluation.

$$S_a = \frac{1}{m_a} \sum_{j \in p_a} | p_{a,j} - v_{a,j} | \qquad (5)$$

In Equation (5), $p_{a,j}$ is the predicted preference, $v_{a,j}$ the real preference, and m_a the number of items that have been evaluated by the new user.

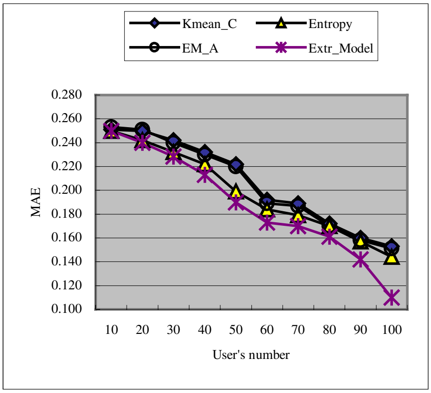
The RSM of an item in a ranked list is determined by user evaluation or user visits. RSM is measured under the premise that the probability of choosing an item lower in the list decreases exponentially. Suppose that each item is put in a decreasing order of value *j*, based on the weight of user preference. Equation (6) calculates the expected utility of user U_a□s RSM on the ranked item list.

$$R_a = \sum_j \frac{\max(V_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \qquad (6)$$

In Equation (6), *d* is the mid-average value of the item, and α is its halflife. The halflife is the number of items in a list that has a 50/50 chance of either review or visit. In the evaluation phase of this paper the halflife value of 5 shall be used. In Equation (7), the RSM is used to measure the accuracy of predictions about the new user.

$$R = 100 \times \frac{\sum_u R_u}{\sum_u R_u^{\max}} \qquad (7)$$

In Equation (7), if the user has evaluated or visited an item that ranks highly in a ranked list, $R_u^{\max}$ is the maximum expected utility of the RSM. For evaluation, this paper uses all of the following methods: the proposed method using a model user (Extr_Model), the method of recommendation using K-means user clustering (K-means_C) [11], the method of recommendation using feature of group(Entropy) [15], the method using EM algorithm(EM_A)[26]. These methods are compared by changing the number of clustered users. Also, the proposed method was compared with the previous method using memory based collaborative filtering technique(Pearson_C) [10] by changing the number of user evaluations on items. Fig. 3 and Fig. 4 show the MAE and RSM of Extr_Model, Kmeans_C, Entropy, and EM_A based on Equation (5) and Equation (7). In Fig. 3 and Fig. 4, as the number of users increases, the performance of the Extr_Model increases, whereas Kmeans_C, EM_A, and Entropy show no notable change in performance.



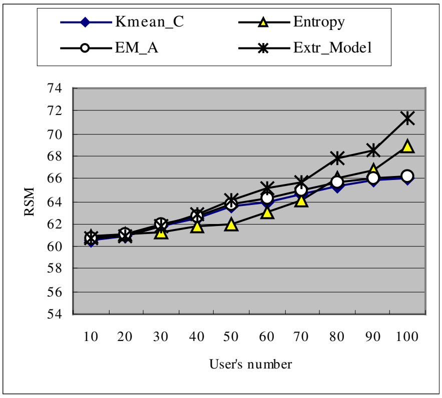
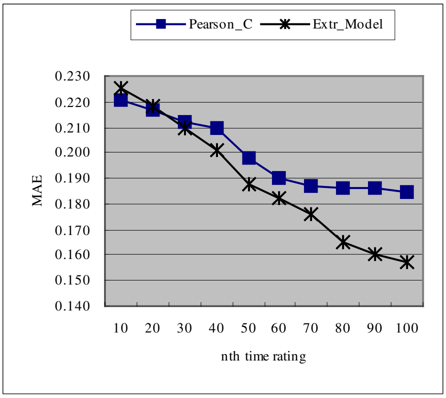


Fig. 3. MAE varying at the number of users

Fig. 4. Rank scoring varying at the number of users

In terms of accuracy of prediction, it is evident that method Extr_Model is more superior to others. On the other side, in case that the number of users is small, the performance of Extr_Model decreases a little. We must study this problem in the future. Fig. 5 and Fig. 6 show the MAE and RSM of Extr_Model and Pearson_C when the number of user□s evaluations is increased. In Fig. 5 and Fig. 6 the Extr_Model outperforms the other in accuracy.



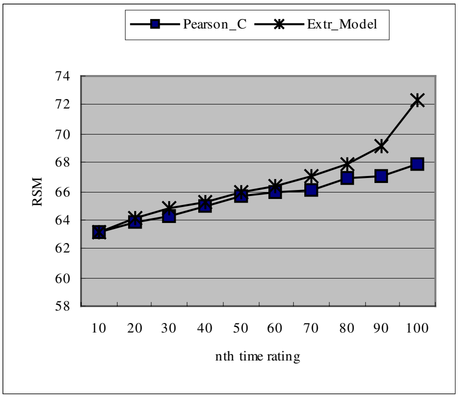


Fig. 5. MAE at n□th rating

Fig. 6. RSM at n□th rating

5 Conclusions

The proposed method in this paper enabled dynamic recommendation because it decreased the inaccuracy of recommendations based on unproven user preferences by optimizing users in collaborative filtering database. Entropy was used to address the collaborative filtering system□s shortcomings whereby items were recommended according to the degree of correlation of the two most similar users within a group. The Na□ve Bayes classifier classified items based on their semantic relations, and then the genetic algorithm clustered users into groups based on the classified items with the reduced dimension. Therefore, the combination of the Na□ve Bayes with the genetic algorithm has proved that the problem of sparsity could be solved without reducing the accuracy of recommendation. Moreover, the proposed method in this paper also reduced the time for retrieving the most similar users within the group by predicting the typical preferences. As the result, this method made dynamic recommendation possible.

In the future, if the proposed method is used with personalized learning agent recommendation systems, the accuracy will, of course, be enhanced. In addition, we are looking forward to testing this proposed method in broader environments and to compare it with other similar systems, as well as improve the accuracy of the clustering and recommendations.

Reference

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[2] K. Alsabti, S. Ranka, and V. Singh, "An Efficient K-Means Clustering Algorithm," http://www.cise.ufl.edu/ranka/, 1997.

[3] Bansal, N., Blum, A., Chawla, S., "Correlation clustering," Proceedings of the 43rd Annual Symposium on Foundations of Computer Science , 238-247 , 2002.

[4] Billsus, D., and Pazzani, M. J., "Learning Collaborative Information Filters," In Proc. Of ICML, pp. 46-53, 1998.

[5] John. S. Breese and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proceedings of the Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.

[6] Goldberg, D., Nichols, D., Oki, B. B., and Terry, D., "Using Collaborative Filtering to Weave an Information Tapestry," Communications of the ACM., 1992.

[7] Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., "Combining Collaborative Filgering With Personal Agents for Better Recommendations," In Proc. Of the AAAI-'99, pp. 439-446, 1999.

[8] Hill, W., Stead, L., Rosenstein, M., and Furnas, G., "Virtual Community of Use," In Proc. of CHI'95, 1995.

[9] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J., "GroupLens:Applying Collaborative Filtering to Usenet News," Communicatins of the ACM, Vol. 40, No. 3, pp. 77-87, 1997.

[10] A. Kohrs and B. Merialdo, "USING CATEGORY-BASED COLLABORATIVE FILTERING IN THE ACTIVE WEBMUSEUM," Proceedings of the IEEE International Conference on Multimedia and Expo - Vol. 1 , 2000.

[11] Taek-Hun Kim, Young-Suk Ryu, Seok-In Park, and Sung-Bong Yang, “An Improved Recommendation Algorithm in Collaborative Filtering,” EC-Web 2002, pp. 254-261, 2002.

[12] M. Gordon, "Probabilistic and genetic algorithms for document retrieval," Communication of the ACM,31, pp. 1208-1218, 1988.

[13] S-J. Ko and J-H. Lee, “Feature Selection using Association Word Mining for Classification,” In Proceedings of the Conference on DEXA2001, LNCS2113, pp. 211-220, 2001.

[14] S-J. Ko, J-H. Choi, and L-H. Lee, “Bayesian Web Document Classification through Optimizing Association Word,” In Proceedings of IEA/AIE2003, LNCS, 2003.

[15] Y. S. Lee and S. W. Lee, “Group Feature Selection using Entropy Weight and SVD,” Transaction of KISS(B), Vol. 29, No. 4, 2002.

[16] G. J. McLachlan and T. Krishnan, The EM Algorithm and Extensions, New York: John Wiley and Sons, 1997.

[17] T. Michael, Maching Learning, McGraw-Hill, pp. 154-200, 1997.

[18] M. Pazzani, D. Billsus, Learning and Revising User Profiles: The Identification of Interesting Web Sites, Machine Learning, Kluwer Academic Publishers, pp. 313-331, 1997.

[19] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., “GroupLens:An Open Architecture for Collabroative Filtering of Newtnews,” Proc. Of CSCW’94, Chapel Hill, NC, 1994.

[20] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., “Application of Dimensionality Reduction in Recommender System-A Case Study,” In ACM WebKDD 200 Web Mining for E-Commerce Workshop, 2000.

[21] Sarwar, B., M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., “Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System,” In Proc. of CSCW’98, 1998.

[22] Badrul Sarwar, George Karypis, Josephp Konstan, and John Ridedl, “Analysis of Recommendation Algorithms for E-Commerce,” Proc. Of The ACM E-Commerce 2000, 2000.

[23] Schmidhuber, J., “Evolutionary computation versus reinforcement learning,” Proceedings of the 2000 26th Annual Conference of the IEEE Industrial Electronics Society, Vol. 4, No. 4, 2000.

[24] Shardanand, U., and Maes, P., “Social Information Filtering: Algorithms for Automating ‘Word of Mouth,” In Proc. of CHI’95, 1995.

[25] I. Soboroff and C. Nicholas, “Combining content and collaboration in text filtering,” In Proceedings of the IJCAI’99 Workshop on Machine Learning in Information filtering, pp. 86-91, 1999.

[26] L. H. Ungar and D. P. Foster, “Clustering Methods for Collaborative Filtering,” AAAI Workshop on Recommendation Systems, 1998.

Author Index