

Sio-long Ao · Oscar Castillo
Xu Huang
Editors

Intelligent Control and Computer Engineering

Lecture Notes in Electrical Engineering

Volume 70

For other titles published in this series, go to
www.springer.com/series/7818

Sio-Iong Ao • Oscar Castillo • Xu Huang
Editors

Intelligent Control and Computer Engineering

 Springer

Editors

Sio-Iong Ao
International Association of Engineers
Hung To Road 37-39
Hong Kong, Unit 1, 1/F
People's Republic of China
publication@iaeng.org

Xu Huang
University of Canberra
Fac. Information Science & Engineering
Canberra, Aust. Capital Terr.
Australia
xu.huang@canberra.edu.au

Oscar Castillo
Tijuana Institute of Technology
Computer Science
Tijuana
Mexico
publication@iaeng.org

ISSN 1876-1100
ISBN 978-94-007-0285-1
DOI 10.1007/978-94-007-0286-8
Springer Dordrecht Heidelberg London New York

e-ISSN 1876-1119
e-ISBN 978-94-007-0286-8

© Springer Science+Business Media B.V. 2011

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Cover design: VTEX, Vilnius

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

A large international conference on Advances in Intelligent Control and Computer Engineering was held in Hong Kong, March 17–19, 2010, under the auspices of the International MultiConference of Engineers and Computer Scientists (IMECS 2010). The IMECS is organized by the International Association of Engineers (IAENG). IAENG is a non-profit international association for the engineers and the computer scientists, which was founded in 1968 and has been undergoing rapid expansions in recent years. The IMECS conferences have served as excellent venues for the engineering community to meet with each other and to exchange ideas. Moreover, IMECS continues to strike a balance between theoretical and application development. The conference committees have been formed with over two hundred and fifty members who are mainly research center heads, deans, department heads (chairs), professors, and research scientists from over thirty countries. The conference participants are also truly international with a high level of representation from many countries. The responses for the conference have been excellent. In 2010, we received more than one thousand manuscripts, and after a thorough peer review process 56.26% of the papers were accepted (<http://www.iaeng.org/IMECS2010>).

This volume contains 25 revised and extended research articles written by prominent researchers participating in the conference. Topics covered include artificial intelligence, control engineering, decision supporting systems, automated planning, automation systems, systems identification, modelling and simulation, communication systems, signal processing, and industrial applications. The book offers the state of the art of tremendous advances in intelligent control and computer engineering and also serves as an excellent reference text for researchers and graduate students, working on intelligent control and computer engineering.

Sio-Iong Ao
Oscar Castillo
Xu Huang

Contents

Intelligent Control of Reduced-Order Closed Quantum Computation Systems Using Neural Estimation and LMI Transformation	1
Anas N. Al-Rabadi	
Optimal Guidance and Control for Space Robot Operation	15
Takuro Kobayashi and Shinichi Tsuda	
The Application of Genetic Algorithms in Designing Fuzzy Logic Controllers for Plastic Extruders	25
Ismail Yusuf, Nur Iksan, and Nanna Suryana Herman	
Automatic Weight Selection and Fixed-Structure Cascade Controller for a Quadratic Boost Converter	39
Somyot Kaitwanidvilai and Pitsanu Srithongchai	
Availability Studies and Solutions for Wheeled Mobile Robots	47
Adrian Korodi and Toma L. Dragomir	
The Use of Higher-Order Spectrum for Fault Quantification of Industrial Electric Motors	59
Juggrapong Treetrong	
A Newly Cooperative PSO – Multiple Particle Swarm Optimizers with Diversive Curiosity, MPSOα/DC	69
Hong Zhang	
Predicting the Toxicity of Chemical Compounds Using GPTIPS: A Free Genetic Programming Toolbox for MATLAB	83
Dominic P. Searson, David E. Leahy, and Mark J. Willis	
Diversity-Driven Self-adaptation in Evolutionary Algorithms	95
Fanchao Zeng, James Decraene, Malcolm Yoke Hean Low, Suiping Zhou, and Wentong Cai	

A New Rearrangement Plan for Freight Cars in a Train	107
Yoichi Hirashima	
Coevolving Negotiation Strategies for P-S-Optimizing Agents	119
Jeonghwan Gwak and Kwang Mong Sim	
Policy Gradient Approach for Learning of Soccer Player Agents	137
Harukazu Igarashi, Hitoshi Fukuoka, and Seiji Ishihara	
Genetic Algorithm for Forming Buyer Coalition with Bundles of Items in E-Marketplaces	149
Anon Sukstrienwong	
Inside Virtual CIM	163
Ning Zhou, Sev Naglingam, Ke Xing, and Grier Lin	
Supreme Court Sentences Retrieval Using Thai Law Ontology	177
Tanapon Tantisripreecha and Nuanwan Soonthornphisaj	
Genetic Algorithm Based Model for Effective Document Retrieval	191
Hazra Imran and Aditi Sharan	
An Agent-Based Cloud Service Discovery System that Consults a Cloud Ontology	203
Taekgyeong Han and Kwang Mong Sim	
Possible Applications of Navigation Tools in Tilings of Hyperbolic Spaces	217
Maurice Margenstern	
Graph Pattern Matching with Expressive Outerplanar Graph Patterns .	231
Hitoshi Yamasaki, Takashi Yamada, and Takayoshi Shoudai	
Setvectors – An Efficient Method to Predict Cache Contention	245
Michael Zwick	
New Material Model for Describing Large Deformation of Pressure Sensitive Adhesive	259
Kazuhisa Maeda, Shigenobu Okazawa, and Koji Nishiguchi	
QoS Provisioning in EPON Systems with Interleaved Two Phase Polling-Based DBA	271
I-Shyan Hwang, Jhong-Yue Lee, and Zen-Der Shyu	
The Game of n-Player Shove and Its Complexity	285
Alessandro Cincotti	

Modeling the Vestibular Nucleus 293
Alexandru Codrean, Adrian Korodi, Toma-Leonida Dragomir, and Vlad Ceregan

SPECT Lung Delineation 307
Alex Wang and Hong Yan

Intelligent Control of Reduced-Order Closed Quantum Computation Systems Using Neural Estimation and LMI Transformation

Anas N. Al-Rabadi

Abstract A new method of intelligent control for closed quantum computation time-independent systems is introduced. The introduced method uses recurrent supervised neural computing to identify certain parameters of the transformed system matrix $[\tilde{\mathbf{A}}]$. Linear matrix inequality (LMI) is then used to determine the permutation matrix $[\mathbf{P}]$ so that a complete system transformation $\{[\tilde{\mathbf{B}}], [\tilde{\mathbf{C}}], [\tilde{\mathbf{D}}]\}$ is achieved. The transformed model is then reduced using singular perturbation and state feedback control is implemented to enhance system performance. In quantum computation and mechanics, a closed system is an isolated system that can't exchange energy or matter with its environment and doesn't interact with other quantum systems. In contrast to an open quantum system, a closed quantum system obeys the unitary evolution and thus is information lossless that implies state reversibility. The experimental simulations show that the new hierarchical control simplifies the model of the quantum computing system and thus uses a simpler controller that produces the desired performance enhancement and system response.

Keywords Linear matrix inequality · Model reduction · Quantum computation · Recurrent supervised neural computing · State feedback control system

1 Introduction

Due to the fact that current dense hardware implementations are heading towards the critical atomic threshold, quantum computing will rapidly occupy an increasingly important position in building nano-size, super-fast, and ultra-low power consuming systems [1–3, 6, 8, 12]. Other motivations for implementing circuits and systems using quantum computing would include items such as: (1) *power* where

A.N. Al-Rabadi (✉)

The University of Jordan, Faculty of Engineering & Technology, Computer Engineering Department, Amman, Jordan 11942
e-mail: alrabadi@yahoo.com

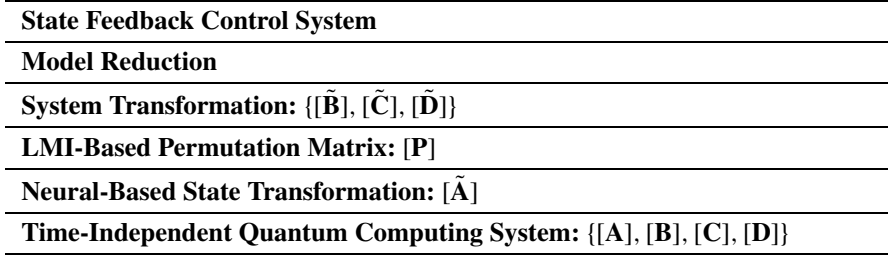


Fig. 1 The introduced control methodology utilized for closed quantum computing systems

the internal computations in quantum computing systems consume no power and only power is consumed when reading and writing operations are performed [1, 6, 8, 12]; (2) *size* where, at the atomic dimension, quantum mechanical effects have to be accounted for; and (3) *speed* where if the properties of superposition and entanglement of quantum mechanics can be usefully employed in the design of circuits and systems, significant computational speed enhancements can be expected [1, 6, 12]. Figure 1 illustrates the layer layout of the introduced closed-system quantum computing control methodology.

2 Fundamentals

This section presents important background on quantum computing systems, supervised neural networks, linear matrix inequality, and model order reduction that will be used later in Sects. 3, 4 and 5.

2.1 Quantum Computation

Quantum computing is an efficient method of computation that uses the dynamic process which is governed by the Schrödinger equation [1, 6, 12]. The one-dimensional time-dependent Schrödinger equation (TDSE) is as follows [1, 5, 6, 12]:

$$-\frac{(\hbar/2\pi)^2}{2m} \frac{\partial^2 |\psi\rangle}{\partial x^2} + V|\psi\rangle = i(\hbar/2\pi) \frac{\partial |\psi\rangle}{\partial t} \quad (1)$$

$$\text{or } H|\psi\rangle = i(\hbar/2\pi) \frac{\partial |\psi\rangle}{\partial t} \quad (2)$$

where \hbar is Planck constant ($6.626 \cdot 10^{-34} \text{ J} \cdot \text{s} = 4.136 \cdot 10^{-15} \text{ eV} \cdot \text{s}$), $V(x, t)$ is the applied potential, m is the particle mass, i is the imaginary number, $|\psi(x, t)\rangle$ is the quantum state, H is the Hamiltonian operator where $H = -[(\hbar/2\pi)^2/2m]\nabla^2 + V$, and ∇^2 is the Laplacian operator.

A general solution to the TDSE is the expansion of a stationary (i.e., time-independent for spatial) basis functions (i.e., eigen states) $U_e(\vec{r})$ using time-dependent (i.e., temporal) expansion coefficients $c_e(t)$ as follows:

$$\Psi(\vec{r}, t) = \sum_{e=0}^n c_e(t) u_e(\vec{r})$$

The expansion coefficients $c_e(t)$ are a scaled complex exponentials as follows:

$$c_e(t) = k_e e^{-i \frac{E_e}{(\hbar/2\pi)} t}$$

where E_e are the energy levels. In quantum computing, the time-independent Schrödinger equation (TISE) is normally used [1, 12]:

$$\nabla^2 \psi = \frac{2m}{(\hbar/2\pi)^2} (V - E) \psi \quad (3)$$

where the solution $|\psi\rangle$ is an expansion over orthogonal basis states $|\phi_i\rangle$ defined in a linear complex vector space called Hilbert space \mathbf{H} as:

$$|\psi\rangle = \sum_i c_i |\phi_i\rangle \quad (4)$$

where the coefficients c_i are called probability amplitudes and $|c_i|^2$ is the probability that the quantum state $|\psi\rangle$ will collapse into the (eigen) state $|\phi_i\rangle$. The probability is equal to the inner product $|\langle \phi_i | \psi \rangle|^2$, with the unitary condition $\sum |c_i|^2 = 1$. In quantum computing, a linear and unitary operator \mathfrak{U} is used to transform an input vector of quantum bits (qubits) into an output vector of qubits. In the two-valued quantum computing, the qubit is a vector of bits which is defined as follows [1, 12]:

$$\text{qubit}_0 \equiv |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{qubit}_1 \equiv |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5)$$

A two-valued quantum state $|\psi\rangle$ is a superposition of quantum basis states $|\phi_i\rangle$. Thus, for the orthonormal computational basis states $\{|0\rangle, |1\rangle\}$, one has the following quantum state:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (6)$$

where $\alpha\alpha^* = |\alpha|^2 = p_0 \equiv$ the probability of having state $|\psi\rangle$ in state $|0\rangle$, $\beta\beta^* = |\beta|^2 = p_1 \equiv$ the probability of having state $|\psi\rangle$ in state $|1\rangle$, and $|\alpha|^2 + |\beta|^2 = 1$. The calculation in quantum computing for multiple systems follows the tensor product (\otimes). For example, given the quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$, one has:

$$\begin{aligned} |\psi_1 \psi_2\rangle &= |\psi_1\rangle \otimes |\psi_2\rangle \\ &= (\alpha_1 |0\rangle + \beta_1 |1\rangle) \otimes (\alpha_2 |0\rangle + \beta_2 |1\rangle) \\ &= \alpha_1 \alpha_2 |00\rangle + \alpha_1 \beta_2 |01\rangle + \beta_1 \alpha_2 |10\rangle + \beta_1 \beta_2 |11\rangle \end{aligned} \quad (7)$$

A physical system (e.g., the hydrogen atom) that is described by the following equation:

$$|\psi\rangle = c_1 |\text{Spinup}\rangle + c_2 |\text{Spindown}\rangle \quad (8)$$

can be used to physically implement a two-valued quantum computing. Another common alternative form of Eq. 8 is as follows:

$$|\psi\rangle = c_1 \left| +\frac{1}{2} \right\rangle + c_2 \left| -\frac{1}{2} \right\rangle \quad (9)$$

Many-valued quantum computing can also be performed. For the three-valued case, the qubit becomes a 3D vector quantum discrete digit (qudit), and in general, for an m -valued quantum computing the qudit is of dimension “many” [1, 12]. For example, one has for the 3-state case, the following qudits:

$$\text{qudit}_0 \equiv |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \text{qudit}_1 \equiv |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{qudit}_2 \equiv |2\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (10)$$

A three-valued quantum state is a superposition of three quantum orthonormal basis states (vectors). Thus, for the orthonormal computational basis states $\{|0\rangle, |1\rangle, |2\rangle\}$, one has the following quantum state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle$$

where $\alpha\alpha^* = |\alpha|^2 = p_0 \equiv$ the probability of having state $|\psi\rangle$ in state $|0\rangle$, $\beta\beta^* = |\beta|^2 = p_1 \equiv$ the probability of having state $|\psi\rangle$ in state $|1\rangle$, $\gamma\gamma^* = |\gamma|^2 = p_2 \equiv$ the probability of having state $|\psi\rangle$ in state $|2\rangle$, and $|\alpha|^2 + |\beta|^2 + |\gamma|^2 = 1$.

The calculation in quantum computing for m -valued multiple systems follow the tensor product in a manner similar to the one demonstrated for the higher-dimensional qubit in the two-valued quantum computing. Several quantum computing systems were used to implement quantum gates from which complete quantum circuits and systems were constructed [1, 6, 12], where several of the two-valued and m -valued quantum circuit implementations use the two-valued and m -valued quantum *Swap*-based and *Not*-based gates [1, 12].

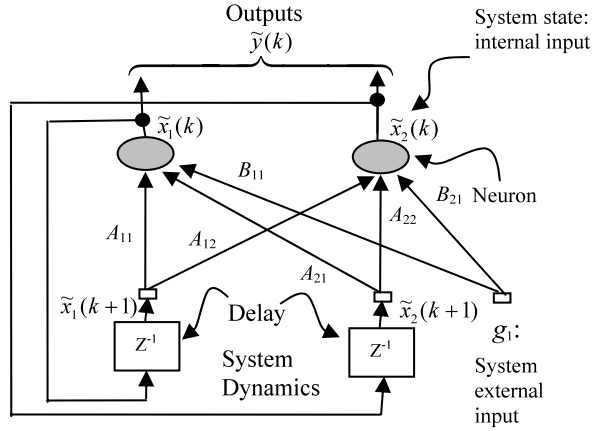
In general, for an m -valued logic, a quantum state is a superposition of m quantum orthonormal basis states (i.e., vectors). Thus, for the orthonormal computational basis states $\{|0\rangle, |1\rangle, \dots, |m-1\rangle\}$, one has the quantum state:

$$|\psi\rangle = \sum_{k=0}^{m-1} c_k |q\rangle_k \quad (11)$$

where $\sum_{k=0}^{m-1} c_k c_k^* = \sum_{k=0}^{m-1} |c_k|^2 = 1$. The calculation in quantum computing for m -valued multiple systems is done similar to the case for the two-valued system.

In quantum mechanical systems, a closed system is an isolated system that doesn't exchange energy or matter with its environment (i.e., doesn't dissipate power) and doesn't interact with other quantum systems. While an open quantum system interacts with its environment and thus dissipates power which results in a non-unitary evolution producing *information loss*, a closed quantum system doesn't exchange energy or matter with its environment and therefore doesn't dissipate power which results in a unitary evolution (i.e., unitary matrix) and thus it is *information lossless*.

Fig. 2 The utilized second order recurrent neural network architecture, where the estimated matrices are given by $\{\tilde{A}_d = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tilde{B}_d = \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}\}$ and $W = [\tilde{A}_d \mid \tilde{B}_d \mid w]$



2.2 Recurrent Supervised Neural Computations

The supervised recurrent neural network which is used for the estimation in this research is based on an approximation of the method of steepest descent [2, 9]. The network tries to match the output of certain neurons to the desired values of the system output at a specific instant of time. Figure 2 shows a network consisting of a total of N neurons with M external input connections for a 2nd order system with two neurons and one external input, where the variable $\mathbf{g}(k)$ denotes the $(M \times 1)$ external input vector which is applied to the network at discrete time k and the variable $\mathbf{y}(k + 1)$ denotes the corresponding $(N \times 1)$ vector of individual neuron outputs produced one step later at time $(k + 1)$.

The derivation of the recurrent algorithm can be started by using $d_j(k)$ to denote the desired (i.e., target) response of neuron j at time k , and $\zeta(k)$ to denote the set of neurons that are chosen to provide externally reachable outputs. A time-varying $(N \times 1)$ error vector $\mathbf{e}(k)$ is defined whose j^{th} element is given by the following relationship:

$$e_j(k) = \begin{cases} d_j(k) - y_j(k), & \text{if } j \in \zeta(k) \\ 0, & \text{otherwise} \end{cases}$$

The objective is to minimize the cost function E_{total} which is obtained by $E_{\text{total}} = \sum_k E(k)$ where $E(k) = \frac{1}{2} \sum_{j \in \zeta} e_j^2(k)$. The dynamical system is described by the following triply indexed set of variables $(\pi_{m\ell}^j)$:

$$\pi_{m\ell}^j(k) = \frac{\partial y_j(k)}{\partial w_{m\ell}(k)}$$

where for every time step k and all appropriate j , m and ℓ , system dynamics are controlled by:

$$\pi_{m\ell}^j(k+1) = \dot{\varphi}(v_j(k)) \left[\sum_{i \in \beta} w_{ji}(k) \pi_{m\ell}^i(k) + \delta_{mj} u_\ell(k) \right]$$

with $\pi_{m\ell}^j(0) = 0$. The values of $\pi_{m\ell}^j(k)$ and the error signal $e_j(k)$ are used to compute the corresponding weight changes with a learning rate (η):

$$\Delta w_{m\ell}(k) = \eta \sum_{j \in \mathcal{C}} e_j(k) \pi_{m\ell}^j(k) \quad (12)$$

Using the weight changes, the updated weight $w_{m\ell}(k+1)$ is calculated as:

$$w_{m\ell}(k+1) = w_{m\ell}(k) + \Delta w_{m\ell}(k) \quad (13)$$

and repeating this computation procedure provides the minimization of the cost function and the objective is therefore achieved.

2.3 Transformation via Linear Matrix Inequality

In this sub-section, the detailed illustration of system transformation using LMI optimization will be presented [2]. Consider the system:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (14)$$

$$y(t) = Cx(t) + Du(t) \quad (15)$$

In order to determine the transformed $[A]$ matrix, which is $[\tilde{A}]$, the discrete zero input response is obtained. This is achieved by providing the system with some initial state values and setting the system input to zero (i.e., $u(k) = 0$). Hence, the discrete system of Eqs. 14, 15, with the initial condition $x(0) = x_0$, becomes:

$$x(k+1) = A_d x(k) \quad (16)$$

$$y(k) = x(k) \quad (17)$$

We need $x(k)$ as a neural network target to train the network to obtain the needed parameters in $[\tilde{A}_d]$ such that the system output will be the same for $[A_d]$ and $[\tilde{A}_d]$. Hence, simulating this system provides the state response corresponding to their initial values with only the $[A_d]$ matrix is being used. Once the input-output data is obtained, transforming the $[A_d]$ matrix is achieved using the neural network training, as will be explained in Sect. 3. The estimated transformed $[A_d]$ matrix is then converted back into the continuous form which yields:

$$\tilde{A} = \begin{bmatrix} A_r & A_c \\ 0 & A_o \end{bmatrix} \quad (18)$$

Having the $[A]$ and $[\tilde{A}]$ matrices, the permutation $[P]$ matrix is determined using the LMI optimization technique [2, 4] as will be illustrated in later sections. The complete system transformation can be achieved by assuming that $\tilde{x} = P^{-1}x$ and then the system of Eqs. 14, 15 can be re-written as follows:

$$P\dot{\tilde{x}}(t) = AP\tilde{x}(t) + Bu(t), \quad \tilde{y}(t) = CP\tilde{x}(t) + Du(t)$$

where ($\tilde{y}(t) = y(t)$). Pre-multiplying the first equation above by $[\mathbf{P}^{-1}]$, one obtains $\{P^{-1}P\dot{\tilde{x}}(t) = P^{-1}AP\tilde{x}(t) + P^{-1}Bu(t), \tilde{y}(t) = CP\tilde{x}(t) + Du(t)\}$ which yields the following transformed model:

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t) \quad (19)$$

$$\tilde{y}(t) = \tilde{C}\tilde{x}(t) + \tilde{D}u(t) \quad (20)$$

where the transformed system matrices are given by:

$$\tilde{A} = P^{-1}AP \quad (21)$$

$$\tilde{B} = P^{-1}B \quad (22)$$

$$\tilde{C} = CP \quad (23)$$

$$\tilde{D} = D \quad (24)$$

Transforming the system matrix $[\mathbf{A}]$ into the form shown in Eq. 18 can be achieved based on the property of matrix reducibility [2, 10].

2.4 Singular Perturbation for Model Order Reduction

Linear time-invariant models of many systems have fast and slow dynamics which is referred to as singularly perturbed systems [2, 11]. Neglecting the fast dynamics of a singularly perturbed system provides a reduced slow model leading to simpler controllers based on the reduced model information [2, 11]. For reduced system formulation, consider the following singularly perturbed system:

$$\dot{x}(t) = A_{11}x(t) + A_{12}\xi(t) + B_1u(t), \quad x(0) = x_0 \quad (25)$$

$$\varepsilon\dot{\xi}(t) = A_{21}x(t) + A_{22}\xi(t) + B_2u(t), \quad \xi(0) = \xi_0 \quad (26)$$

$$y(t) = C_1x(t) + C_2\xi(t) \quad (27)$$

where $x \in \mathfrak{R}^{m_1}$ and $\xi \in \mathfrak{R}^{m_2}$ are the slow and fast state variables, respectively, $u \in \mathfrak{R}^{m_1}$ and $y \in \mathfrak{R}^{m_2}$ are the input and output vectors, respectively, $\{[\mathbf{A}_{ii}], [\mathbf{B}_i], [\mathbf{C}_i]\}$ are constant matrices of appropriate dimensions with $i \in \{1, 2\}$, and ε is a small positive constant. The singularly perturbed system in Eqs. 25, 26, 27 is simplified for $\varepsilon = 0$. By doing the above step, one neglects the system fast dynamics assuming that the state variables ξ have reached the quasi-steady state. Setting $\varepsilon = 0$ in Eq. 26 and assuming $[A_{22}]$ is nonsingular, produces:

$$\xi(t) = -A_{22}^{-1}A_{21}x_r(t) - A_{22}^{-1}B_2u(t) \quad (28)$$

where the index r denotes remained (or reduced) model. By substituting Eq. 28 into Eqs. 25, 26, 27, one obtains the following reduced order model:

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t) \quad (29)$$

$$y(t) = C_r x_r(t) + D_r u(t) \quad (30)$$

for $\{A_r = A_{11} - A_{12}A_{22}^{-1}A_{21}, B_r = B_1 - A_{12}A_{22}^{-1}B_2, C_r = C_1 - C_2A_{22}^{-1}A_{21}, D_r = -C_2A_{22}^{-1}B_2\}$.

3 Neural Estimation with Linear Matrix Inequality-Based Transformation for Closed Reduced-Order Quantum Computation Systems

In this work, it is our objective to search for a similarity transformation that can be utilized within the context of closed time-independent quantum computing systems to decouple a pre-selected eigenvalue set from the system matrix $[\mathbf{A}]$. To achieve this objective, training the neural network to estimate the transformed discrete system matrix $[\tilde{\mathbf{A}}_d]$ is performed [2]. For the system of Eqs. 25, 26, 27, the discrete model of the quantum computing system is obtained as:

$$x(k+1) = A_d x(k) + B_d u(k) \quad (31)$$

$$y(k) = C_d x(k) + D_d u(k) \quad (32)$$

The estimated discrete model of Eqs. 31, 32 can be re-written as:

$$\begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} u(k) \quad (33)$$

$$\tilde{y}(k) = \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} \quad (34)$$

where k is the time index, and the matrix elements of Eqs. 33, 34 were shown in Fig. 2. The recurrent neural network that was presented in Sect. 2.2 can be summarized by defining Λ as the set of indices (i) for which $g_i(k)$ is an external input, which is one external input in the quantum computing system, and by defining β as the set of indices (i) for which $y_i(k)$ is an internal input (or a neuron output), which is two internal inputs (i.e., two system states) in the quantum computing system. Also, we define $u_i(k)$ as the combination of the internal and external inputs for which $i \in \beta \cup \Lambda$. By using this setting, training the network depends on the internal activity of each neuron which is given by the following equation:

$$v_j(k) = \sum_{i \in \Lambda \cup \beta} w_{ji}(k) u_i(k) \quad (35)$$

where w_{ji} is the weight representing an element in the system matrix or input matrix for $j \in \beta$ and $i \in \beta \cup \Lambda$ such that $W = [[\tilde{\mathbf{A}}_d] \quad [\tilde{\mathbf{B}}_d]]$. At the next time step ($k+1$), the output (i.e., internal input) of the neuron j is computed by passing the activity through the nonlinearity $\phi(\cdot)$ as follows:

$$x_j(k+1) = \phi(v_j(k)) \quad (36)$$

With these equations, based on an approximation of the method of steepest descent, the network estimates the system matrix $[\tilde{\mathbf{A}}_d]$ as was shown in Eq. 16 for zero input response. That is, an error can be obtained by matching a true state output with a neuron output as follows:

$$e_j(k) = x_j(k) - \tilde{x}_j(k)$$

The objective is to minimize the cost function $E_{\text{total}} = \sum_k E(k)$ where $E(k) = \frac{1}{2} \sum_{j \in \varsigma} e_j^2(k)$ and ς denotes the set of indices j for the output of the neuron structure. This cost function is minimized by estimating the instantaneous gradient of $E(k)$ with respect to the weight matrix $[\mathbf{W}]$ and then updating $[\mathbf{W}]$ in the negative direction of this gradient. In detailed steps, this may be proceeded as follows:

- Initialize the weights $[\mathbf{W}]$ by a set of uniformly distributed random numbers. Starting at the instant $k = 0$, use Eqs. 35, 36 to compute the output values of the N neurons (where $N = \beta$).
- For each time step k and all $j \in \beta$, $m \in \beta$, and $\ell \in \beta \cup \Lambda$, compute the dynamics of the system governed by the triply indexed set of variables:

$$\pi_{m\ell}^j(k+1) = \dot{\varphi}(v_j(k)) \left[\sum_{i \in \beta} w_{ji}(k) \pi_{m\ell}^i(k) + \delta_{mj} u_\ell(k) \right]$$

with initial conditions $\pi_{m\ell}^j(0) = 0$ and $\delta_{m\ell}$ is given by $(\partial w_{ji}(k)/\partial w_{m\ell}(k))$, which is equal to “1” only when $j = m$ and $i = \ell$ otherwise it is “0”. Note that for the special case of a sigmoidal nonlinearity in the form of a logistic function, the derivative $\dot{\varphi}(\cdot)$ is given by $\dot{\varphi}(v_j(k)) = y_j(k+1)[1 - y_j(k+1)]$.

- Compute the weight changes correspond to the error and system dynamics:

$$\Delta w_{m\ell}(k) = \eta \sum_{j \in \varsigma} e_j(k) \pi_{m\ell}^j(k) \quad (37)$$

- Update the weights in accordance with:

$$w_{m\ell}(k+1) = w_{m\ell}(k) + \Delta w_{m\ell}(k) \quad (38)$$

- Repeat the computation until the desired estimation is achieved.

As was illustrated in Eqs. 16, 17, for the purpose of estimating only the transformed system matrix $[\tilde{\mathbf{A}}]$, the training is based on the zero input response. Once the training is complete, the obtained weight matrix $[\mathbf{W}]$ is the discrete estimated transformed system matrix. Transforming the estimated system back to the continuous form yields the desired continuous transformed system matrix $[\tilde{\mathbf{A}}]$. Using the LMI optimization technique that was illustrated in Sect. 2.3, the permutation matrix $[\mathbf{P}]$ is determined. Hence, a complete system transformation, as was shown in Eqs. 19, 20, is achieved. To perform the order reduction, the system in Eqs. 19, 20 are written as:

$$\begin{bmatrix} \dot{\tilde{x}}_r(t) \\ \dot{\tilde{x}}_o(t) \end{bmatrix} = \begin{bmatrix} A_r & A_c \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \tilde{x}_r(t) \\ \tilde{x}_o(t) \end{bmatrix} + \begin{bmatrix} B_r \\ B_o \end{bmatrix} u(t) \quad (39)$$

$$\begin{bmatrix} \tilde{y}_r(t) \\ \tilde{y}_o(t) \end{bmatrix} = \begin{bmatrix} C_r & C_o \end{bmatrix} \begin{bmatrix} \tilde{x}_r(t) \\ \tilde{x}_o(t) \end{bmatrix} + \begin{bmatrix} D_r \\ D_o \end{bmatrix} u(t) \quad (40)$$

where the system transformation enables us to decouple the original system into retained (r) and omitted (o) eigenvalues. The retained eigenvalues are the dominant eigenvalues that produce slow dynamics and the omitted eigenvalues are the non-dominant eigenvalues that produce fast dynamics. Equation 39 can be re-written as $\{\dot{\tilde{x}}_r(t) = A_r \tilde{x}_r(t) + A_c \tilde{x}_o(t) + B_r u(t), \dot{\tilde{x}}_o(t) = A_o \tilde{x}_o(t) + B_o u(t)\}$.

The coupling term $A_c \tilde{x}_o(t)$ maybe compensated for by solving for $\tilde{x}_o(t)$ in the second equation above by setting $\dot{\tilde{x}}_o(t)$ to zero using the singular perturbation method (by setting $\varepsilon = 0$). Doing so, the following is obtained:

$$\tilde{x}_o(t) = -A_o^{-1} B_o u(t) \quad (41)$$

Using $\tilde{x}_o(t)$, we get the reduced model given by:

$$\dot{\tilde{x}}_r(t) = A_r \tilde{x}_r(t) + [-A_c A_o^{-1} B_o + B_r] u(t) \quad (42)$$

$$y(t) = C_r \tilde{x}_r(t) + [-C_o A_o^{-1} B_o + D] u(t) \quad (43)$$

Therefore, the overall reduced order model is:

$$\dot{\tilde{x}}_r(t) = A_{or} \tilde{x}_r(t) + B_{or} u(t) \quad (44)$$

$$y(t) = C_{or} \tilde{x}_r(t) + D_{or} u(t) \quad (45)$$

where the details of the overall reduced matrices $\{[A_{or}], [B_{or}], [C_{or}], [D_{or}]\}$ are shown in Eqs. 42, 43.

4 Model Order Reduction of the Quantum Computation Systems Using Neural Estimation and Linear Matrix Inequality Transformation

Let us implement the time-independent quantum computing closed-system using the particle in finite-walled box potential V for the general case of m -valued quantum computing in which the resulting distinct energy states are used as the orthonormal basis states [2]. The dynamical TISE of the one-dimensional particle in finite-walled box potential V is expressed as follows:

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{2m}{(\hbar/2\pi)^2} (E - V) \Psi = 0$$

which also can be re-written as $\frac{\partial^2 \Psi}{\partial x^2} = \frac{2m}{\hbar^2} (V - E) \Psi$, where m is the particle mass, and $\hbar = (\hbar/2\pi)$ is the reduced Planck constant (which is also called the Dirac constant) $\cong 1.055 \cdot 10^{-34} \text{ J} \cdot \text{s} = 6.582 \cdot 10^{-16} \text{ eV} \cdot \text{s}$. Thus, for $\{x_1 = \Psi, x_2 = \frac{\partial \Psi}{\partial x}, x'_1 = x_2, x'_2 = \frac{\partial^2 \Psi}{\partial x^2}\}$, the state space model of the time-independent closed quantum computing system is given as:

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{2m(V-E)}{\hbar^2} & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} u \quad (46)$$

$$y = (1 \quad 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (0) u \quad (47)$$

For simulation reasons, Eqs. 46, 47 can also be re-written equivalently as:

$$\begin{pmatrix} -x'_2 \\ x'_1 \end{pmatrix} = \begin{bmatrix} 0 & \frac{2m(E-V)}{\hbar^2} \\ -1 & 0 \end{bmatrix} \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} u \quad (48)$$

$$y = (0 \quad 1) \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix} + (0)u \quad (49)$$

Also, for conducting the simulations, one may often need to scale the system Eq. 48 without changing the system dynamics. Thus, by scaling both sides of Eq. 48 by a scaling factor a , the following set of equations is obtained:

$$a \begin{pmatrix} -x'_2 \\ x'_1 \end{pmatrix} = a \begin{bmatrix} 0 & \frac{2m(E-V)}{\hbar^2} \\ -1 & 0 \end{bmatrix} \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} u \quad (50)$$

$$y = (0 \quad 1) \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix} + (0)u \quad (51)$$

Therefore, one obtains the following set of quantum system matrices:

$$A = a \begin{bmatrix} 0 & \frac{2m(E-V)}{\hbar^2} \\ -1 & 0 \end{bmatrix} \quad (52)$$

$$B = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (53)$$

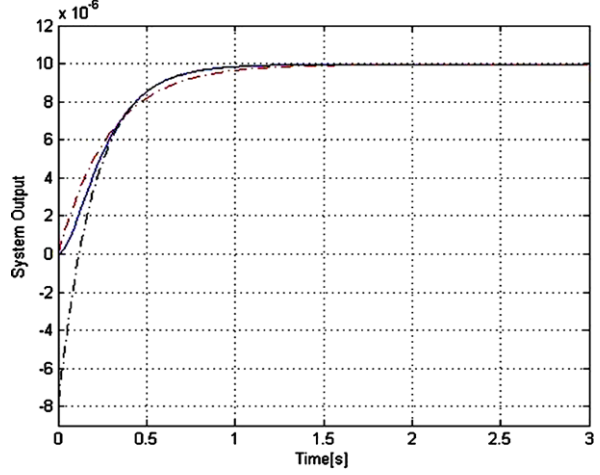
$$C = [0 \quad 1] \quad (54)$$

$$D = [0] \quad (55)$$

The specifications of the system matrix in Eq. 52 for the particle in finite-walled box are determined by (1) potential box width L (in nanometer), (2) particle mass m , and (3) the potential value V (i.e., potential height in electron Volt). As an example, consider the particle in a finite-walled potential with specifications of $(E - V) = 88$ MeV and a very light particle with a particle mass of $N = 10^{-33}$ of the electron mass (where the electron mass $m_e \cong 9.109 \cdot 10^{-27}$ g = $5.684 \cdot 10^{-12}$ eV/(m/s)²). This system was discretized using the sampling rate $T_s = 0.005$ second and simulated for a zero input. Hence, based on the obtained simulated output data and by using NN to estimate the subsystem matrix $[A_c]$ of Eq. 18 with a learning rate $\eta = 0.015$, the transformed system matrix $[\tilde{A}]$ was obtained where $[A_r]$ is set to provide the dominant eigenvalues (i.e., slow dynamics) and $[A_o]$ is set to provide the non-dominant eigenvalues (i.e., fast dynamics) of the original system. Thus, when training the system, the second state $\tilde{x}_o(t)$ of the transformed model in Eq. 39 is unchanged due to the restriction of $[0 \quad A_o]$ seen in $[\tilde{A}]$. This may lead to an undesired starting of the system response, but fast system overall convergence.

Using $[\tilde{A}]$ along with $[A]$, the LMI is implemented to obtain $\{[\tilde{B}], [\tilde{C}], [\tilde{D}]\}$ which makes a complete model transformation. Then, by using singular perturbation for model reduction, the reduced order model is obtained. Thus, by implementing the previously stated system specifications and by using the squared reduced Planck constant of $\hbar^2 = 43.324 \cdot 10^{-32}$ (eV · s)², one obtains the following scaled system matrix from Eq. 52:

Fig. 3 (Color online) Input-to-output quantum computing system step responses: full-order system model (*solid blue line*), transformed reduced-order model (*dashed black line*), and non-transformed reduced-order model (*dashed red line*)



$$\begin{aligned}
 a^{-1}A &= \begin{bmatrix} 0 & \frac{2m(E-V)}{\hbar^2} \\ -1 & 0 \end{bmatrix} \cong \begin{bmatrix} 0 & 2.32 \cdot 10^{-6} \\ -0.95 & 0.003 \end{bmatrix} \\
 &= \begin{pmatrix} -1 \\ 5000 \end{pmatrix} \begin{bmatrix} 0 & -0.0116 \\ 4761.9 & -16 \end{bmatrix}
 \end{aligned}$$

Accordingly, the eigenvalues were found to be $\{-5.0399, -10.9601\}$. For a step input, simulating the original and transformed reduced order models along with the non-transformed reduced order model produced the results shown in Fig. 3.

5 The Design of State Feedback Controller for the Reduced-Order Closed Quantum Models

In this research, since the closed quantum computing system is a 2nd order system reduced to a 1st order, we will investigate the system stability and enhancing performance by implementing the simple method of the s -domain pole replacement [2, 7]. For the reduced model in Eqs. 44, 45, a state feedback controller can be designed. For example, this can be achieved by replacing the system eigenvalues with new faster eigenvalues. Hence, let the control input be:

$$u(t) = -K\tilde{x}_r(t) + r(t) \quad (56)$$

where K is to be designed based on the desired system eigenvalues.

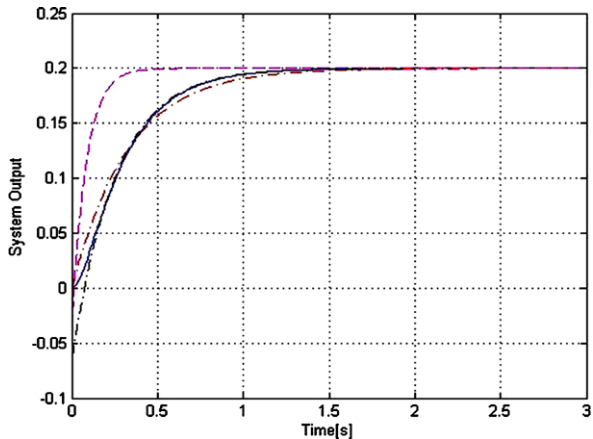
Replacing the control input $u(t)$ in Eqs. 44, 45 by the above new control input in Eq. 56 yields the following reduced system:

$$\dot{\tilde{x}}_r(t) = A_{or}\tilde{x}_r(t) + B_{or}[-K\tilde{x}_r(t) + r(t)] \quad (57)$$

$$y(t) = C_{or}\tilde{x}_r(t) + D_{or}[-K\tilde{x}_r(t) + r(t)] \quad (58)$$

which can be re-written as:

Fig. 4 (Color online) Enhanced system step responses using pole placement; full-order system model (*solid blue line*), transformed reduced model (*dashed black line*), non-transformed reduced model (*dashed red line*), and the controlled transformed reduced (*dashed pink line*)



$$\begin{aligned}\dot{\tilde{x}}_r(t) &= A_{or}\tilde{x}_r(t) - B_{or}K\tilde{x}_r(t) + B_{or}r(t) \rightarrow \dot{\tilde{x}}_r(t) \\ &= [A_{or} - B_{or}K]\tilde{x}_r(t) + B_{or}r(t) \\ y(t) &= C_{or}\tilde{x}_r(t) - D_{or}K\tilde{x}_r(t) + D_{or}r(t) \rightarrow y(t) \\ &= [C_{or} - D_{or}K]\tilde{x}_r(t) + D_{or}r(t)\end{aligned}$$

The overall closed-loop model is then written as:

$$\dot{\tilde{x}}(t) = A_{cl}\tilde{x}_r(t) + B_{cl}r(t) \quad (59)$$

$$y(t) = C_{cl}\tilde{x}_r(t) + D_{cl}r(t) \quad (60)$$

such that the closed-loop system matrix $[A_{cl}]$ will provide the new desired eigenvalues. As an example, consider the following non-scaled quantum system:

$$A = \begin{bmatrix} 0 & -0.385 \\ 142.857 & -18 \end{bmatrix}, \quad B = \begin{bmatrix} 0.077 \\ 0 \end{bmatrix}, \quad C = [0 \quad 1], \quad D = [0]$$

Using the transformation-based reduction technique, one obtains the reduced model $\{\dot{\tilde{x}}_r(t) = [-3.901]\tilde{x}_r(t) + [-5.255]u(t), y_r(t) = [-0.197]\tilde{x}_r(t) + [-0.066]u(t)\}$ with the eigenvalue of -3.901 . Now, suppose that a new eigenvalue $\lambda = -12$ that will produce faster system dynamics is desired for this reduced model. This objective is achieved by first setting the desired characteristic equation as $\lambda + 12 = 0$. To determine the feedback control gain K , the characteristic equation is accordingly utilized by using Eqs. 57–60 which yields $\{(\lambda\mathbf{I} - A_{cl}) = 0 \rightarrow \lambda\mathbf{I} - [A_{or} - B_{or}K] = 0\}$ after which the feedback control gain K is calculated to be -1.5413 , and the closed-loop system now has the eigenvalue of -12 . Simulating the reduced model using a sampling rate $T_s = 0.005$ second and a learning rate $\eta = 0.015$ with the new eigenvalue for the same original system input (i.e., step input) has generated the response in Fig. 4.

6 Conclusions and Future Work

A new method of intelligent control via neural estimation and LMI-based transformation for controlling time-independent quantum computing systems is implemented, and a simple state feedback control using pole placement was then applied on the reduced quantum computing model that achieved the required system response. Future work will investigate the implementation of the introduced hierarchical control onto other quantum systems such as the non-linear, relativistic, and time-dependent quantum computing systems.

References

1. Al-Rabadi, A.N.: *Reversible Logic Synthesis: From Fundamentals to Quantum Computing*. Springer, Berlin (2004)
2. Al-Rabadi, A.N.: Recurrent supervised neural computation and LMI model transformation for order reduction-based control of linear time-independent closed quantum computing systems. In: *Lecture Notes in Engineering and Computer Science: Proc. of the Int. MultiConference of Engineers and Computer Scientists, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 911–923 (2010)
3. Bennett, C.H., Landauer, R.: The fundamental physical limits of computation. *Sci. Am.* **253**(1), 48–56 (Jul 1985)
4. Boyd, S., El-Ghaoui, L., Feron, E., Balakrishnan, V.: *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia (1994)
5. Dirac, P.: *The Principles of Quantum Mechanics*. Oxford University Press, London (1930)
6. Feynman, R.: Quantum mechanical computers. *Opt. News* **11**, 11–20 (1985)
7. Franklin, G., Powell, J., Emami-Naeini, A.: *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading (1994)
8. Fredkin, E., Toffoli, T.: Conservative logic. *Int. J. Theor. Phys.* **21**, 219–253 (1982)
9. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Macmillan College, New York (1994)
10. Horn, R., Johnson, C.: *Matrix Analysis*. Cambridge University Press, Cambridge (1985)
11. Kokotovic, P., O'Malley, R., Sannuti, P.: Singular perturbation and order reduction in control theory – an overview. *Automatica* **12**(2), 123–132 (1976)
12. Nielsen, M., Chuang, I.: *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge (2000)

Optimal Guidance and Control for Space Robot Operation

Takuro Kobayashi and Shinichi Tsuda

Abstract This paper deals with a control of space robot for capturing moving targets. It would be desirable to use the space robot to repair the failed satellite and to remove space debris since the work load to do these tasks by astronauts will be extremely heavy. Extensive studies have been done for the control of space robot. Unfortunately these studies have not incorporated the orbital motion which is essential for space robot. Coplanar motion between space robot and target is discussed in this study. Suboptimal control, which uses piecewise optimized feedback gain by optimal tracking control method, is applied to chase the target. Also Hill's equation was applied to the relative orbital equations of motions. Based on the above formulation dynamical simulation was conducted to demonstrate the validity of our approach.

Keywords Hill's equation · Motion control · Optimal control · Space robot

1 Introduction

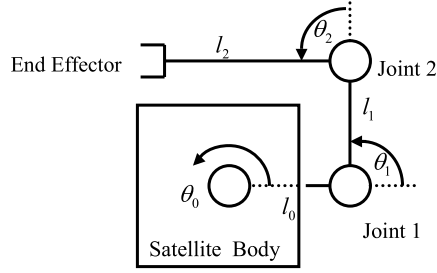
Recently space robots have been extensively used for the space activities like the International Space Station. Moreover Japanese government announced future space programs which will utilize the robot technology like for Moon exploration.

A lot of studies have also been made for more advanced space robots like a robot satellite, in which the robot will be operated in an autonomous manner. These robots are expected to play an important role, like a space debris capture and retrieval. However these studies have not given any consideration about the effect of orbital motion, which generates the relative motion between the space robot and moving target.

T. Kobayashi (✉)

The course of Aerospace, School of Engineering, Tokai University, 1117 Kitakaname Hiratsuka, Kanagawa, 259-1292 Japan
e-mail: 9amjm005@mail.tokai-u.jp

Fig. 1 Model of space robot with end effector



Based on the above consideration the control of space robot, like a satellite robot, is discussed, in which its hand, an end effector of the space robot, tracks the moving target [1]. Firstly the kinematics of space robot is formulated using the generalized Jacobian [2]. And then the control method to correct the position error between the end effector and the target with feedback is defined. Dynamical equation was also derived to obtain the relation between joint variables and applied torque. In this development a linearized approximation, in which the centrifugal and Coriolis terms were neglected, was done by assuming the small deviation of joint variables and their velocity. The tracking control is formulated by applying optimal control theory, LQR [3]. Piecewise optimization for time-varying state-space equation is applied in this paper. This is a practical solution for suboptimal control and as shown in the simulation result it looks working well. Hill's equation was introduced to deal with the relative motion between the space robot and the target, which are assumed to be on a circular orbit without loss of generality.

2 Model of Space Robot

Figure 1 shows the model of space robot with an end effector.

Two dimensional motion is assumed here, therefore, θ_0 gives the satellite attitude angle and two joint angles are defined to express the robot arm posture.

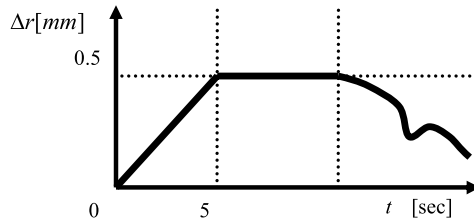
The relation between the position of the end effector and joint variables is given by the following general equation:

$$\mathbf{r} = \mathbf{f}(\mathbf{q}_M) \quad (1)$$

where \mathbf{r} and \mathbf{q}_M are the end effector position and joint variable vectors, respectively. The velocity relation between end effector and the joint variables is as following:

$$\dot{\mathbf{r}} = \mathbf{J}_M \dot{\mathbf{q}}_M \quad (2)$$

Since the base of the space robot is not fixed, the satellite attitude is also changed by the arm operation. In this respect the generalized Jacobian was proposed to analytically deal with this issue. Momentum and angular momentum conservations give us the relations between end effector and joint variable velocity as below:

Fig. 2 Profile of Δr 

$$\dot{\mathbf{r}} = \mathbf{J}^* \dot{\mathbf{q}}_{\mathbf{M}} \quad (3)$$

$$\mathbf{J}^* = \hat{\mathbf{J}}_{\mathbf{M}} - \hat{\mathbf{J}}_{\mathbf{S}} \hat{\mathbf{I}}_{\mathbf{S}}^{-1} \hat{\mathbf{I}}_{\mathbf{M}} \quad (4)$$

$$\hat{\mathbf{I}}_{\mathbf{S}} \dot{\mathbf{q}}_{\mathbf{S}} + \hat{\mathbf{I}}_{\mathbf{M}} \dot{\mathbf{q}}_{\mathbf{M}} = 0 \quad (5)$$

where $\hat{\mathbf{J}}_{\mathbf{M}}$: Jacobian matrix of the robot arm, $\hat{\mathbf{J}}_{\mathbf{S}}$: Jacobian matrix of the satellite, $\hat{\mathbf{I}}_{\mathbf{M}}$: robot arm inertia tensor and $\hat{\mathbf{I}}_{\mathbf{S}}$: satellite inertia tensor.

From (4) and (5) we obtain the following relationships:

$$\Delta \mathbf{q}_{\mathbf{M}} = \mathbf{J}^{*-1} \Delta \mathbf{r} \quad (6)$$

$$\Delta \mathbf{q}_{\mathbf{S}} = -\hat{\mathbf{I}}_{\mathbf{S}}^{-1} (\hat{\mathbf{I}}_{\mathbf{M}} \Delta \mathbf{q}_{\mathbf{M}}) \quad (7)$$

Thus when we define $\Delta \mathbf{r}$, we can obtain $\Delta \mathbf{q}_{\mathbf{M}}$ and $\Delta \mathbf{q}_{\mathbf{S}}$.

$\Delta \mathbf{r}$ will be given at a small interval. This idea is shown in Fig. 2 as an example, in which

$$\Delta \mathbf{r} = \frac{\mathbf{r}_t - \mathbf{r}_d}{|\mathbf{r}_t - \mathbf{r}_d|} \Delta r \quad (8)$$

$$\Delta r = \begin{cases} 0.1 \cdot t & 0 \leq t \leq 5 \\ 0.5 & 5 < t \\ |\mathbf{r}_t - \mathbf{r}_d| & |\mathbf{r}_t - \mathbf{r}_d| < 0.5 \end{cases} \quad (9)$$

in order to eliminate the error of the end effector (6) was modified as follows:

$$\Delta \mathbf{q}_{\mathbf{M}} = \mathbf{J}^{*-1} [\Delta \mathbf{r} + \lambda (\mathbf{r}_d - \mathbf{r})] \quad (10)$$

where λ , \mathbf{r}_d and \mathbf{r} are feedback gain, goal position and current position, respectively.

3 Dynamics of Space Robot

The dynamics of the robot is generally expressed in the following form:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (11)$$

where $\mathbf{M}(\mathbf{q})$: inertia matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$: matrix derived from centrifugal and Corioli forces, and $\boldsymbol{\tau}$: applied torque. The linearization was made in the vicinity of the current posture of the robot and the second term in left hand side of the above equation was neglected by assuming both the small angle deviation and very slow joint

velocity. This will be shown in the simulation result. Thus we obtain a linearized dynamical equation as bellow:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\boldsymbol{\tau} \quad (12)$$

The specific expression of this equation is given in Appendix 1.

4 Optimal Tracking Control

Linear optimal control, LQR, is applied to the control of the space robot to track the target. Generally state space representation for optimal tracking control is as follows. The state equation and observation equation are given by the vector equations as below:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \quad (13)$$

and when a desired trajectory $\mathbf{q}_d(t)$ is defined in the time interval between t_0 and t_f , the tracking error is given by the following equation:

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \quad (14)$$

where $\mathbf{q}_d(t) \equiv \mathbf{q}_d(n) = \mathbf{q}_d(n-1) + [\Delta\mathbf{q}_S(t) \ \Delta\mathbf{q}_M(t)]^T$ for the n -th step. And the performance index is defined in the quadratic formula:

$$J = \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (15)$$

where \mathbf{Q} and \mathbf{R} are a positive semi-definite symmetric and a positive definite symmetric matrices, respectively.

The solution x^0 of the above optimal control problem is resolved as the TPBVP (Two Point Boundary Value Problem) expressed by the following differential equation:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{p}} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{C}^T\mathbf{Q}\mathbf{C} & -\mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^T\mathbf{Q} \end{bmatrix} \mathbf{q}_d \end{aligned} \quad (16)$$

where $\mathbf{p}(t)$ satisfies the relation:

$$\mathbf{u}^0(t) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{p}(t) \quad (17)$$

Let us assume that $\mathbf{p}(t)$ is as below:

$$\mathbf{p}(t) = \mathbf{K}(t)\mathbf{x}(t) + \mathbf{p}_1(t) \quad (18)$$

where $\mathbf{K}(t)$ is a symmetric and positive definite matrix with the same dimension as the vector $\mathbf{x}(t)$. Replacing $\mathbf{p}(t)$ by $\mathbf{p}_1(t)$, we obtain the following state equation

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{p}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{F} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{G} & -\mathbf{F}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p}_1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^T\mathbf{Q} \end{bmatrix} \mathbf{q}_d \quad (19)$$

$$\mathbf{F} = \mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} \quad (20)$$

$$-\mathbf{G} = \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} - \mathbf{K}\mathbf{A} - \mathbf{A}^T\mathbf{K} - \mathbf{C}^T\mathbf{Q}\mathbf{C} - \dot{\mathbf{K}} \quad (21)$$

Let us substitute the followings into (19) and (21):

$$\mathbf{G} = \mathbf{0} \quad \text{and} \quad \dot{\mathbf{K}} = \mathbf{0}$$

Then we have the state equation and algebraic Riccati equation as below.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{p}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{F} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ -\mathbf{G} & -\mathbf{F}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p}_1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{C}^T\mathbf{Q} \end{bmatrix} \mathbf{q}_d \quad (22)$$

$$\mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} - \mathbf{K}\mathbf{A} - \mathbf{A}^T\mathbf{K} - \mathbf{C}^T\mathbf{Q}\mathbf{C} = \mathbf{0} \quad (23)$$

We apply a feedback gain \mathbf{K} , which is given by the steady state solution, to our tracking control.

5 Relative Motion by Orbital Motion

The relative motion is described by Hill's equation with the reference coordinate system of the target. X axis is directed toward the target movement and Y axis is normal to the orbital plane. Z axis is reversely directed toward the earth center.

Then we obtain the following equations:

$$\begin{aligned} \ddot{x} &= -g_t \frac{x}{r_t} - 2\omega\dot{z} - \dot{\omega}z + \omega^2x + A_x \\ \ddot{y} &= -g_t \frac{y}{r_t} + A_y \\ \ddot{z} &= 2g_t \frac{z}{r_t} - 2\omega\dot{x} - \dot{\omega}x + \omega^2z + A_z \end{aligned} \quad (24)$$

where g_t is gravity acceleration, and \mathbf{A} is external force, for example, by thrusters. As assumed in the previous section the target is on a circular orbit and external forces are not acting, then $\dot{\omega} = 0$ and $\omega = \sqrt{g_t/r_t}$.

$$\begin{aligned} \ddot{x} + 2\omega\dot{z} &= 0 \\ \ddot{y} + \omega^2y &= 0 \\ \ddot{z} - 2\omega\dot{x} - 3\omega^2z &= 0 \end{aligned} \quad (25)$$

The solution of these equations is shown in Appendix 2. In order to avoid the collision between the satellite and the target, the satellite robot arm will be approaching from the line of the target movement. And the coordinate system is converted to the space robot reference frame with X axis directed toward the reverse of the target movement and Y axis directed toward the reverse of the center of the earth.

Table 1 Satellite and robot characteristics

	Body	Link 1	Link 2
Mass [kg]	1500	30	50
Link length [m]	1.5	1.5	2.5
Moment of inertia [kg m ²]	1000	22.5	26
Initial angle [deg]	-90	90	90

Table 2 Target orbital properties

	Target
Center of rotation: X direction [m]	4.8
Center of rotation: Y direction [m]	1
Rotational radius [m]	1.5
Altitude [km]	1000
Angular velocity [deg/sec]	1
Relative velocity: X direction [m/sec]	-0.001
Relative velocity: Y direction [m/sec]	-0.0001

6 Simulation Results

The simulation, in which the end effector is tracking the target during 90 seconds, was conducted.

We made the following assumptions for simulation study:

- (1) The target is in reachable zone by the end effector of the robot during the period, and as noted above,
- (2) Only coplanar motions are allowed for the robot and target.

Parameters used in the simulation are shown Tables 1 and 2.

And the following parameters are assumed in the simulation:

$$\lambda = 0.05$$

and

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Figure 3 illustrates the initial geometrical relation between the robot satellite and moving target.

Figures 4 and 5 show the end effector has reached the target with an error less than 0.001 m at 45.6 second. The maximum velocity of the joints was smaller than 0.05 rad/sec which is compatible with other space robot arm.

Figure 6 illustrates the detailed relative distance between end effector and target from 45 second to 90 second. This chart shows good tracking performance.

Fig. 3 Initial geometry between target and satellite

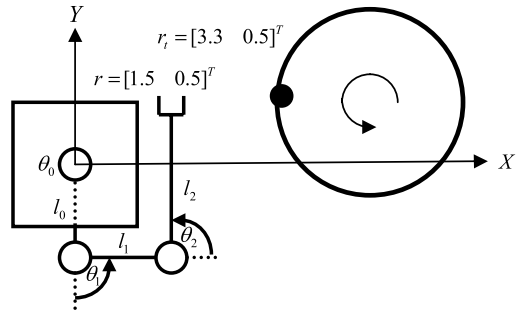


Fig. 4 Positions of target and robot at 0, 30, 60, 90 seconds

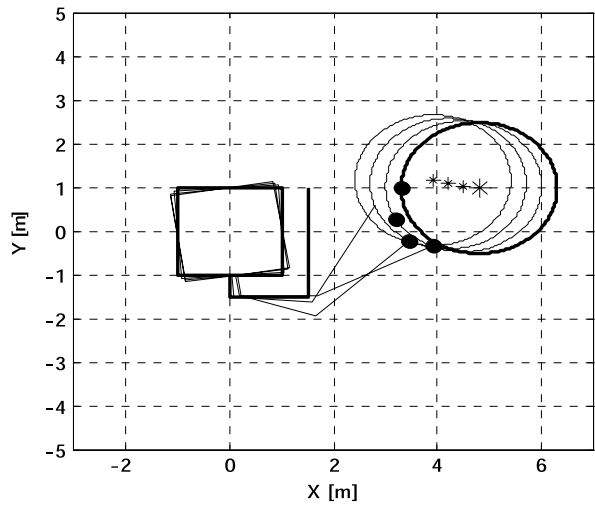
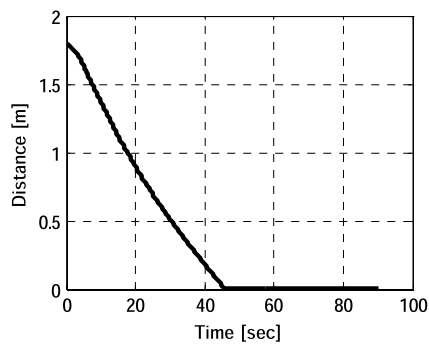


Fig. 5 Relative distance between end effector and target



The histories of joint angles and their velocity, which were obtained from the kinematic equation (10), are given in Figs. 7 and 8. These are trajectories followed by tracking control.

Fig. 6 Relative distance between end effector and target

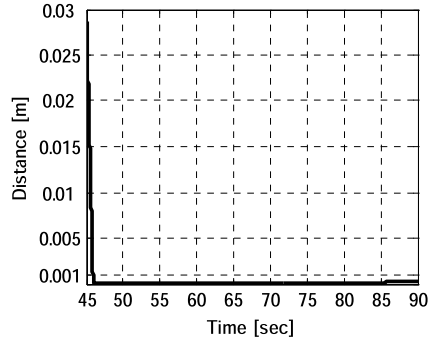


Fig. 7 History of joint angles by kinematics

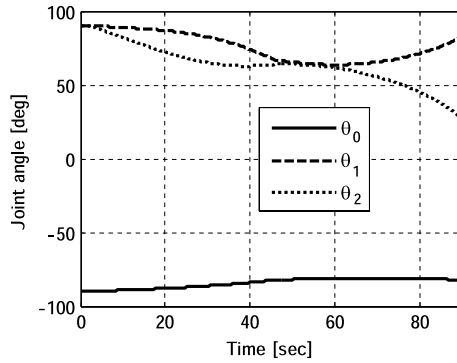
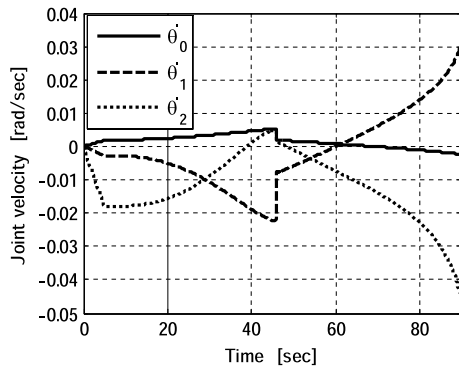


Fig. 8 History of joint angle velocity by kinematics



Figures 9 and 10 illustrate the result of optimal tracking control, in which good agreement with the desired trajectories given by Figs. 9 and 10 is observed. And small angular velocities of joints are preserved.

Fig. 9 Joint angle history by optimal tracking control

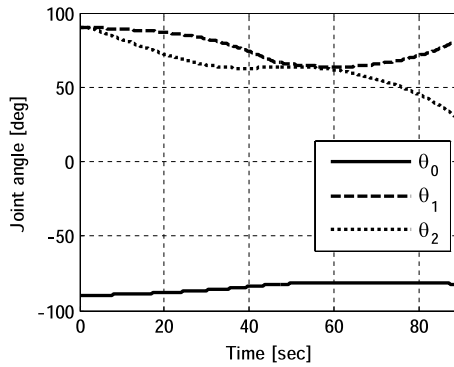
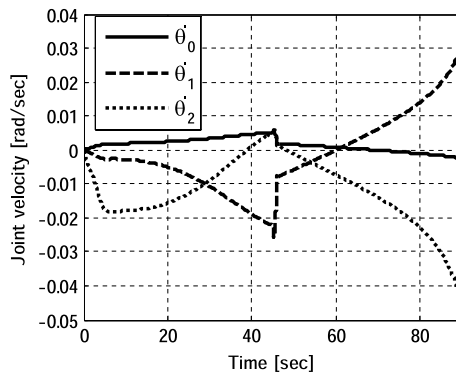


Fig. 10 Joint angle velocities by optimal tracking control



7 Conclusions

The space robot tracking control to capture a target was discussed. In general for orbiting targets like the space debris we have to consider two motions, rotation about their center of mass and orbital motion, at the same time. Especially targets like space debris and failed satellites are noncooperative for capturing them by space robot so that tracking control is inevitable.

Kinematics and dynamics are formulated, including orbital motion which has not been discussed yet. And the optimal tracking control method was applied by using piecewise optimized feedback gains.

Simulation result shows satisfactory performance of the control system by our approach.

Appendix 1: State Space Equation of Joint Variables for Space Robot

$$\frac{d}{dt} \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ M_{11}(\mathbf{q})^{-1} & M_{12}(\mathbf{q})^{-1} & M_{13}(\mathbf{q})^{-1} \\ 0 & 0 & 0 \\ M_{21}(\mathbf{q})^{-1} & M_{22}(\mathbf{q})^{-1} & M_{23}(\mathbf{q})^{-1} \\ 0 & 0 & 0 \\ M_{31}(\mathbf{q})^{-1} & M_{32}(\mathbf{q})^{-1} & M_{33}(\mathbf{q})^{-1} \end{bmatrix} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \end{bmatrix}$$

Appendix 2: Solutions of Hill's Equations

$$x = x_0 - 2\frac{\dot{z}_0}{\omega}(1 - \cos \omega t) + \left(4\frac{\dot{x}_0}{\omega} + 6z_0\right) \sin \omega t - (6\omega z_0 + 3\dot{x}_0)t$$

$$y = \frac{\dot{y}_0}{\omega} \sin \omega t + y_0 \cos \omega t$$

$$z = 4z_0 + 2\frac{\dot{x}_0}{\omega} - \left(2\frac{\dot{x}_0}{\omega} + 3z_0\right) \cos \omega t + \dot{z}_0 \cos \omega t$$

References

1. Kobayashi, T., Tsuda, S.: Control of space robot for moving target capturing. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS2010, Hong Kong, 17–19 March 2010, pp. 946–950 (2010)
2. Umetani, Y., Yoshida, K.: Resolved motion rate control of space robotic manipulators with generalized Jacobian matrix. JRSJ 7(4), 327–337 (1989)
3. Uchida, H., Nonami, K.: Robust control system design for optimal tracking servo-system with trajectory following. SICE 32(8), 1175–1182 (1996)

The Application of Genetic Algorithms in Designing Fuzzy Logic Controllers for Plastic Extruders

Ismail Yusuf, Nur Iksan, and Nanna Suryana
Herman

Abstract This paper investigates the application of Genetic Algorithms (GA) in the design and implementation of Fuzzy Logic Controllers (FLC) for temperature control in an extruder. The importance of FLC is during the process of selecting the membership functions. What is best to determine the membership functions is the first question that has be addressed. It is important therefore to select accurate membership functions but these methods possess one common weakness where conventional FLC use membership functions generated by human operators. In this situation the membership function selection process is done by trial and error and it runs step by step which is too long to arrive at a solution to the problem. This research proposes a method that may help users to determine the membership functions of FLC using GA optimization for the fastest process in solving problems. The data collection is based on simulation results and the results refer to the maximum overshoot. From the results presented, the system arrives at better and more exact results and the value of overshoot is decreased from 1.2800 for FLC without GA, to 1.0011 for FGA.

Keywords Temperature · Extruder · Fuzzy logic · Genetic algorithms · Membership function

1 Introduction

Automatic control has played an important role in the advance of engineering and science. Automatic control is essential in such industrial operations as controlling pressure, temperature, humidity, viscosity, and flow in the process industries. While modern control theory has been easy to practice [14], FLC have been rapidly gaining

I. Yusuf (✉)

Faculty of Information and Communication Technology, Technical University of Malaysia
Malacca (UTeM), 76109 Durian Tunggal, Malaysia
e-mail: ariel_ismail@yahoo.com

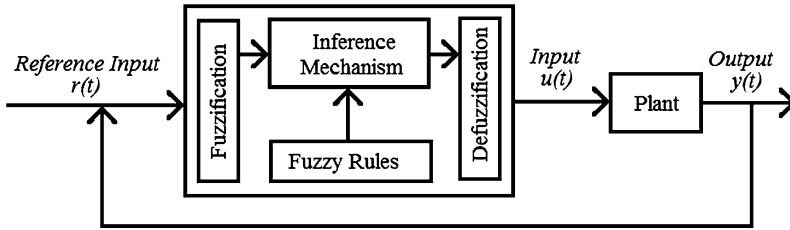


Fig. 1 Fuzzy controller architecture

popularity among practicing engineers. This increase of popularity can be attributed to the fact that fuzzy logic provides a powerful vehicle that allows engineers to incorporate human reasoning in the control algorithm.

In our daily lives from manufacturing plant production lines, medical equipment, agriculture to consumer products such as washing machines and air-conditioners, FLC can be applied. A good example is the controller temperature set for plastic extruders by FLC [21]. When extruding certain materials, the temperatures along the extruder must be accurately controlled in accordance with properties of the extruder and the particular polymer. If the temperatures are not accurately controlled, the molten polymer will not be uniform and may decompose as a result of excessive temperatures.

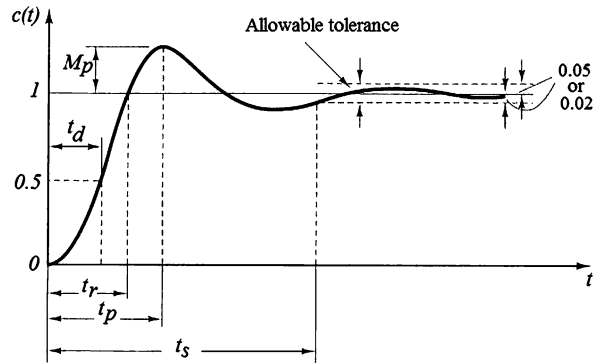
One of the problems associated with prior extruder control systems is the design of the barrel zone temperature controllers [11, 19]. Preferably, these controllers are designed with a high sensitivity to disturbance signals. However when a change in a temperature set point occurs, there is a danger of saturating the zone temperature controllers as the magnitude of the temperature set point changes are generally greater than the magnitude of the disturbances. The sensitivity of the controller to disturbance signals must be reduced to prevent saturation of the controllers to set point changes [1, 11, 19]. It is therefore important to select accurate membership functions for temperature settings in extruder control systems.

Taking the above explanation, we propose to use control systems based on FLC. The process of selecting membership functions is an important part of FLC [9]. With conventional FLC the membership function is generated by human operators manually. This has one common weakness since the selection process is done by trial and error and runs step by step which are too long in solving the problem [1] and interpretation mistakes may happen [18]. A new approach for optimum coding of fuzzy controllers is using GA to determine membership function especially designed for particular situations. We use GA to tune the membership function to terms of each fuzzy variable.

2 Fuzzy Logic Controller

Fuzzy logic process (fuzzy inferences) provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how

Fig. 2 Transient and steady-state response analyses



to control a system. The fuzzy controller is composed of the following four elements: fuzzy rules, inference mechanism, fuzzification and defuzzification interface [15] as shown in Fig. 1.

For the closed loop control system, the signal output is denoted by $y(t)$, the input signal is denoted by $u(t)$, and the reference input to the fuzzy controller is denoted by $r(t)$. The actuating error signal is a difference both of input signal and feedback signal, it will be feeding into the controller to reduce error and bring the output of the system to a desired value [2].

We must have a basis for analyzing and comparing the performance of various control systems. Performance of various control system can be analyzed by concentrating on maximum overshoot response as shown graphically in Fig. 2.

3 Genetic Algorithms

GA borrow ideas from and attempt to simulate Darwin’s theory on natural selection and Mandel’s work in genetic inheritance. The usual form of GA was described by Goldberg [6]. GA are stochastic search techniques based on the mechanism of natural selection and natural genetics. It differs from conventional search techniques. It starts with an initial set of random solutions called “population”. Each individual in the population is called a “chromosome”, representing a solution to the problem at hand. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness [5]. To create the next generation, new chromosomes called offspring are formed by both crossover and mutation operations, and a new generation is formed by selection and rejection [3, 12]. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome which hopefully represents the optimal solution of the problem.

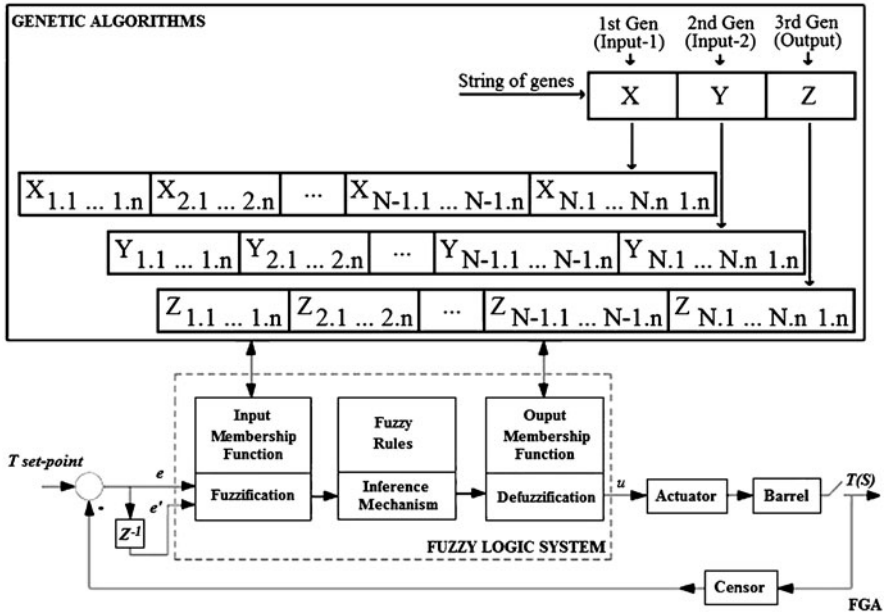


Fig. 3 Block diagram plastic extruder by fuzzy genetic algorithms

4 Materials and Methods

Basically, GA has had a great measure of success in search and optimization problem solving. In this research, GA are used to improve the performance of the fuzzy controller. Considering that the main attribute of the GA is its ability to solve the topological structure of an unknown system, then the problem for determining the fuzzy membership functions can also fall into this category. The concept of using GA for determining membership function was carried out early in a review [20] and was applied [9].

This research uses fuzzy genetic algorithms (FGA) for designing temperature control in an extruder. There are three possible zones in a thermoplastic screw: feed zone, melt zone and pressurizing zone [10, 11]. Each zone will be equipped with one or more thermocouples for temperature control. The pressurizing zone (also called *metering* or *melt conveying*) gives the plastic uniform pressure and flow characteristics. Research [21] was to design a system control based on FLC for controlling the temperature so the desired melting point in the “pressurizing zone” could be maintained. The conceptual idea is to have an automatic and intelligent scheme to tune the fuzzy membership functions of the closed loop control for extruder machine using GA, as proposed in Fig. 3.

A mathematical model of plastic polymer for a single screw extruder is given:

$$T(s) = \frac{(0.0035 \cdot Q_h) + (240 \cdot T_m) + (0.000202 \cdot T_u)}{6(S + 240.000202)} \tag{1}$$

where:

$$\begin{aligned}
 Q_h &= \text{heat input rate (kcal/sec)} \\
 T_m &= \text{temperature of polymers (110}^\circ\text{C)} \\
 T_u &= \text{temperature of outlet air (40}^\circ\text{C)} \\
 T(s) &= \text{transfer function of temperature } t(s)
 \end{aligned}$$

For obtaining final (tuned) membership function by using GA, some functional mapping of the system will be given. Parameters of the initial membership function are generated and coded as real numbers that are concatenation to make one long string to represent the whole parameter set of membership function. A fitness function is used to evaluate the fitness value of each set of membership function. Then the reproduction, crossover and mutation operators are applied to obtain the optimal population (membership function). The final tuned value describes the membership function which is proposed.

Having now learnt the procedures for designing a FLC, the practical realization of this system to determine the membership function in FLC is not an easy process. The dynamic variation fuzzy input of membership functions will be main the stumbling block to this design.

4.1 Data Structures

The most important structures in GA are those that represent genes and chromosomes. Most researchers [1, 4] have represented the chromosome string of genes as a binary code but in our case, the real numbers code will be used as real numbers code is a more natural representation than binary code [17].

This is research intended to determine membership function of the fuzzy partition based on real numbers. Each gene corresponds to one linguistic variable whose definition is what the GA tries to evolve. In fuzzy systems, membership function was representing the value of a linguistic variable and for our simulation design, there are three kinds of linguistic variables: the variable error signal as input-1 is parameter X and the rate of change in error as input-2 is parameter Y and the controller output is parameter Z , as shown in Fig. 4. For every variable there are five shapes of membership functions, three are triangular and two trapezoid. If membership function has triangular form, then it can be described by three parameters. A fixed number of real numbers is used to define each of the three parameters which completely defines the specific triangular membership function. If it is trapezoidal, it requires four parameters.

4.2 Fitness Function

In the evolution of nature, the highest valuable individual fitness will survive whereas the low valuable individual will die. The fitness function is the basis of

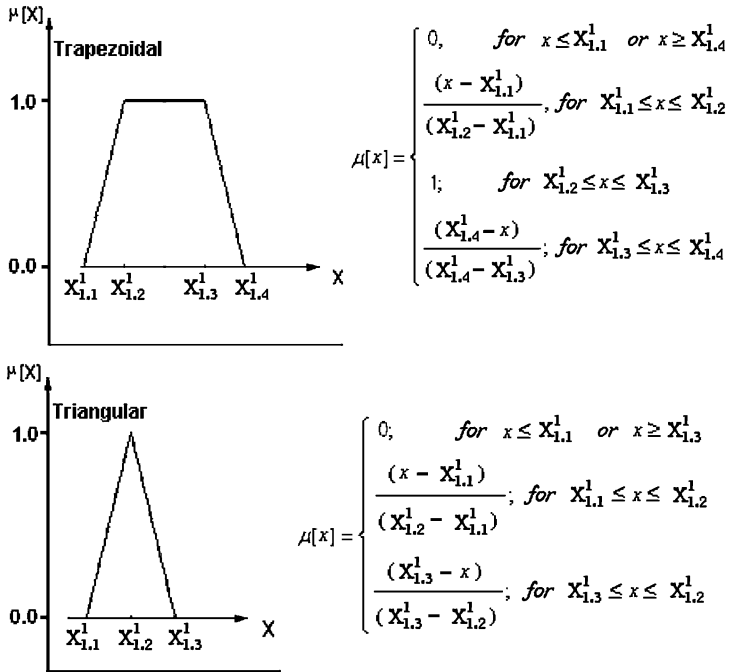


Fig. 4 Structure of chromosome

the survival of the fittest premise of genetic algorithms. It is responsible for evaluating the parameter sets, and choosing which parameter sets are suitable. The most difficult part of the fitness function is to design the function for producing parameters that are reliable and effective. The fuzzy controller operates in a closed-loop specification, which can be analyzed by the maximum overshoot. The maximum overshoot (M_p) is the maximum peak value of the response curve measured from the unity and the amount of the maximum (percent) overshoot directly indicates the relative stability of the system [1, 2, 14, 16].

The fitness function measures how close each individual in the population meets the given specification at a given instant of time. Thus for a given individual in the population, the maximum overshoot is computed respectively by performing a simulation of the closed loop control system with the candidate controller (FGA) and model of the plant.

4.3 Genetic Parameters

The stumbling block for implementing GA is to set the values for the various parameters, such as population size, probability of crossover rate, and probability of mutation rate. These parameters typically interact with one another nonlinearly, so

they cannot be optimized for all situations. This research is set to population sizes of 5, 10 and 100 for comparing, while the probability of crossover rate is 0.1, 0.7 and 0.9 where the probability of mutation rate is 0.001, 0.05 and 0.1.

4.4 Termination Condition

GA will typically run forever until an appropriate termination condition is reached. Termination criteria in the GA literature are somewhat arbitrary because the statistical nature of the search prohibits knowing when the population will evolve to a given fitness value. Typically, GA searches are terminated after [3, 18]:

1. A maximum number of generations or individual population members have been evaluated, or
2. The diversity of the string values in the population has been significantly reduced, or
3. Goal fitness (such as 1.0) has been achieved.

The termination condition for this research is the one that defines the maximum number of generations to be produced. When the generation number is completed by the GA, the new populations generating process is finished, and the best solution is the one among individuals that most adapted to the evaluation function.

5 Result and Analysis

As mentioned previously, the most important aspect in implementing GA for improvement of performance is how to set values of the various parameters, such as population size, the probability of crossover and the mutation rate. These parameters typically interact with each other.

- 1st experiment: combination of population size: (5: 0.7: 0.001), (10: 0.7: 0.001) and (100: 0.7: 0.001).
- 2nd experiment: combination of mutation rate: (10: 0.7: 0.001), (10: 0.7: 0.005) and (10: 0.7: 0.05).
- 3rd experiment: combination of crossover rate: (10: 0.1: 0.001), (10: 0.7: 0.001) and (10: 0.9: 0.001).
- 4th experiment: combination of crossover rate: (10: 0.1: 0.05), (10: 0.7: 0.05) and (10: 0.9: 0.05).

Referring to previous available research [7, 8, 13, 20], it seems (10: 0.7: 0.001) to be the best combination rate and is chosen for the tests although it does not give optimum results in the further tests.

In the first experiment we compare the convergence rates of populations: 5, 10 and 100 individuals while the probability of crossover rate is 0.7 and the probability

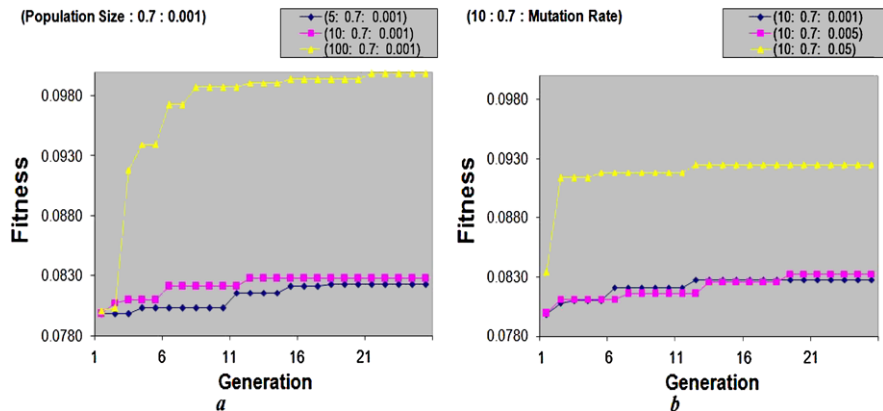


Fig. 5 Comparison of convergent: (a) Population size, (b) mutation rate

of mutation rate is 0.001. All the data can be shown in the graph in Fig. 5a. These results show that the performance of GA for first generation with a population size of 5 and 10 are the same i.e. 0.0798. Then this result will increase in the next generation until final generation. In the training data, we found the fitness 0.0823 at the 25th generation when the population size is 5 and 0.0828 at the 25th generation when the population size is 10. For the best result, the fitness is 0.0999 at 25th generation when the population size is 100. But the consume time varies significantly, from 370 minutes when the population size is 10 and increases to 1205 minutes when the population size is 100. These results show that the performance of GA is very sensitive to the population size.

In the second experiment, we make a comparison of the convergence of mutation: 0.001, 0.005 and 0.05 while the probability of crossover rate is 0.7 and the population size sets to 10 individuals. All the data can be shown in the Fig. 5b. From Fig. 5b, we can see the effect of a probability mutation rate on the fitness value. If the value of mutation rate is high then the fitness value will get better. The highest fitness value is 0.0925 for a probability mutation rate of 0.05, secondly it is 0.0833 (probability mutation rate of 0.005) and the lowest value is 0.0828 (probability mutation rate of 0.001).

If the probability mutation rate is high then the time consumed will increase. From 370 minutes (when the probability of mutation rate is 0.001) it increases to 384 minutes (when the probability of mutation rate is 0.005) and 467 minutes (when the probability of mutation rate is 0.05). This situation shows that GA need a longer time if probability of mutation rate is higher. In this case, the combination of parameters (10: 0.7: 0.05) shows the best performance of GA while the combination parameter (10: 0.7: 0.001) shows the lowest performance of GA. This is different with our hypothesis.

Result of the third experiment can be show in the following graph in Fig. 6a. Figure 6a shows a comparison of convergence of crossover rate: 0.1, 0.7 and 0.9 while the probability of mutation is 0.001 and the population size sets to 10 individuals. In

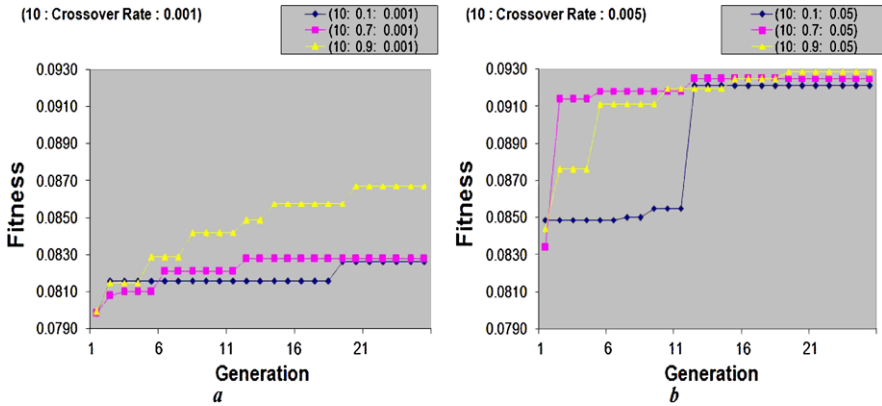


Fig. 6 Comparison of convergent while mutation rate set to: (a) 0.001, and (b) 0.05

this experiment, the combination parameter (10: 0.9: 0.001) shows the best performance of GA, while the combination parameter (10: 0.1: 0.001) shows the lowest performance of GA.

We can now see the effect of probability crossover rate on the fitness value. If the value of crossover rate is high then the fitness value will get better. The highest fitness value is 0.0867 for a probability crossover rate of 0.9, secondly it is 0.0828 (probability crossover rate of 0.7) and the lowest value is 0.0826 (probability crossover rate of 0.1). If the probability crossover rate is high then the time consumed will increase. From 316 minutes (when the probability of crossover rate is 0.1) increases to 370 minutes (when the probability of crossover rate is 0.7) and 425 minutes (when the probability of crossover rate is 0.9). This situation shows that GA need a longer time if probability of mutation rate is higher.

In the fourth experiment, we make a comparison of the convergence rates of crossover rate: 0.1, 0.7 and 0.9 while the probability of mutation is 0.05 and the population size sets to 10 individuals, as shown in Fig. 6b. In this experiment, the highest fitness value is 0.0929 while combination of parameters are set to (10: 0.9: 0.05), the second fitness value is 0.0925 while combination of parameters are set to (10: 0.7: 0.05), and the lowest fitness value is 0.0921 while combination of parameters are set to (10: 0.1: 0.05).

The comparison between Fig. 6a and 6b shows the fitness value only slightly different for any value of crossover rate if the probability mutation rate is set to 0.05 (in Fig. 6b). Figure 6a shows it has significant fitness value if crossover rate sets to 0.9 compared to 0.1 and 0.7. This situation shows that the values of probability crossover rate and mutation rate interact, both will affect each other.

From the results presented in this experiment, the system which we have developed is very helpful to determine membership function for the fastest processing in solving problems. Figure 7 (a, b) shows the membership function existing (without GA) and will be used in initial population in the first chromosome, first population and first generation.

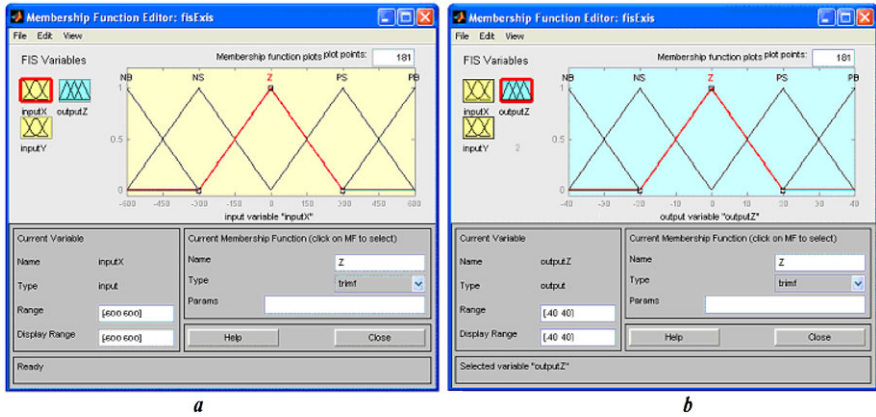


Fig. 7 Existing membership (without GA): (a) 1st input, (b) output

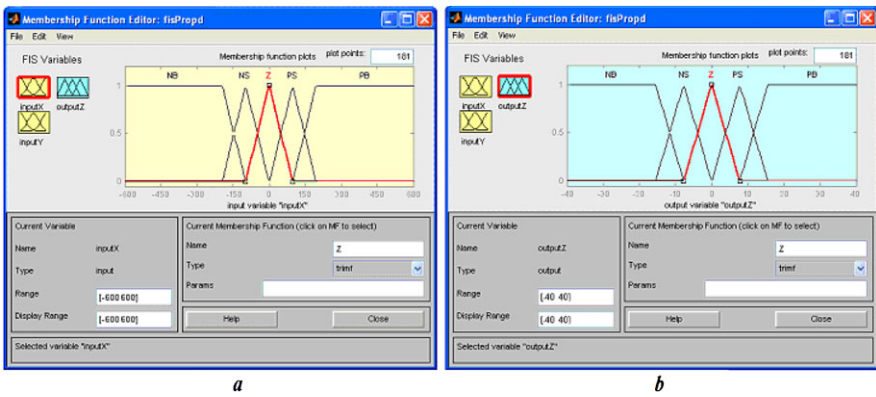


Fig. 8 Proposed membership (with GA/FGA): (a) 1st input, (b) output

Compare with Fig. 8 (a, b), where GA were applied into the fuzzy system to determine the membership function (proposed membership function).

During the execution of FGA, the fitness result has been recorded as data. The membership function evolves after evolution is complete and will be tested using the data for comparing the FLC with and without GA. Generally speaking, after membership function has been tuned by using GA then it will improve the performance of a FLC. We can compare the FLC with and without GA thus the visualization as shown in Fig. 9. After programming is executed then membership function is regulated automatically. A better and more exact result is obtained: the value of overshoot is 1.2800 for FLC without GA (shown in Fig. 9a) and decreases to 1.0011 for FGA (shown in Fig. 9b). It is clear the GA are very promising to improve the performance of a FLC and for getting more accuracy in order to achieve optimum results.

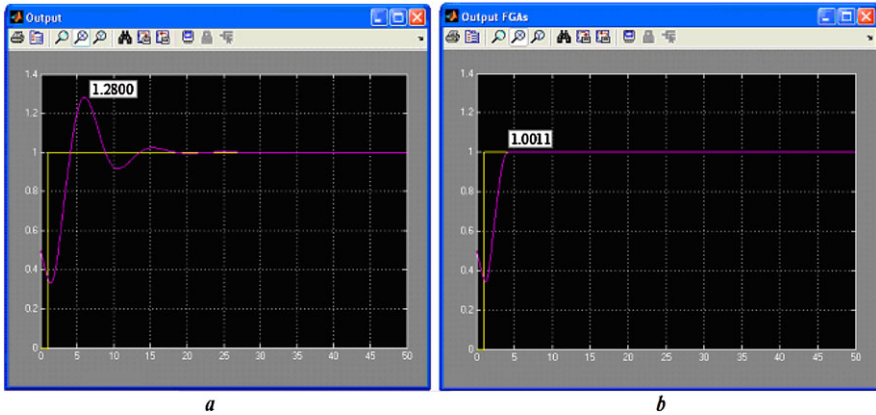


Fig. 9 Response system for membership function: (a) Without GA, (b) with GA/FGA

6 Discussion

GA has been successfully applied to solve optimization problems. In this research, GA are implemented for determining the membership function of FLC in a plastic extruder to control the temperature. By designing compact data structures for genes and chromosomes and an accurate fitness evaluation function, GA have been implemented to effectively find more accurate membership functions for the fuzzy system. The data structures adopted are compact and thus very conveniently manipulated by genetic operators.

Population size is a significant factor to improve the performance of GA. Generally speaking, the larger population size will be better for performance of GA; a larger population size has more diverse chromosomes and thus will contain a large number of good chromosomes. This is helpful to achieve a better solution. But it also increases the time required for the population to converge and it causes longer processing time. The population size specifies how many organisms should be stored in memory at any given time.

The selection operation is the probability of the chromosome (offspring) for surviving and will be used in the further chromosome calculations (crossover and mutation) for the next generation. Mutation serves to create random diversity in the population, while crossover serves as an accelerator that promotes emergent behavior of components. Increasing the mutation probability will increase areas of random search, but it also makes for a loss of the genetic material. Increasing the crossover probability will increase the recombination of chromosomes, but it also disrupts good chromosomes.

This research found several matters which are important factors that influence performance of GA. There are three basic issues:

1. Selection procedure for choosing suitable candidates for mating.
2. The value of α for scaling factor in arithmetic crossover method.

3. The value of b for parameter degree in dynamic (or non-uniform) mutation method.

7 Conclusion and Future Research

The population size was a significant factor to improve the performance of GA. The larger population size will be better for performance of GA, but longer in processing time. GA need a longer time if probability of crossover and mutation is higher. The interaction between probability of crossover and mutation is significant; both of them will affect each other. In this research the following significant features have been identified:

- Combination parameter (10: 0.9: 0.05) will give the best value for solving our problem. The value of fitness is 0.0929 (maximum overshoot = 1.0770) and the processing time is 580 minutes.
- Combination parameter (100: 0.7: 0.001), will give the best fitness value. The value of fitness is 0.0999 (maximum overshoot = 1.0011) and the processing time is 1205 minutes.

The performance of GA can be further improved by using different combinations of selection strategies, crossover and mutation methods and other genetic parameters such as population size, probability of crossover and mutation rate.

References

1. Altinten, A., Erdogan, S., Hapoglu, H., Aliev, F., Albaz, M.: Application of fuzzy control method with genetic algorithm to a polymerization reactor at constant set point. *Inst. Chem. Eng. Trans. IChemE A*, 1012–1018 (2006)
2. Boulet, B.: *Fundamentals of Signal and Systems*. Da Vinci Engineering Press, Hingham (2006)
3. Chen, H.C., Cheng, S.H.: Genetic algorithms based optimization design of a pid controller for an active magnetic bearing. *Int. J. Comput. Sci. Netw. Secur.* **6**(12), 95–99 (2006)
4. Galantucci, L.M., Percoco, G., Spina, R.: Assembly and disassembly planning by using fuzzy logic & genetic algorithms. *Int. J. Adv. Robot. Syst.* **1**(2), 67–74 (2004)
5. Gen, M., Cheng, R.: *Genetic Algorithms & Engineering Design*. John Wiley & Sons, New York (1997), pp. 1–7
6. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley Longman, California (1999), pp. 1–54.
7. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybern.* **SMC-16**, 122–128 (1996)
8. Hayat, Wirth (June 2, 2010), <http://www.journal.au.edu/ijcem/may97/article2.html>
9. Kissi, M., Ramdani, M., Tollabi, M., Zakarya, D.: Determination of fuzzy logic membership function using GA: application to olfaction, pp. 616–619
10. Liptak, B.G.: *Instrument Engineers' Handbook: Process Control and Optimization*, 3rd ed. Butterworth-Heinemann, Stoneham (2006)
11. Lu, C.H., Tsai, C.C.: Adaptive decoupling predictive temperature control for an extrusion barrel in a plastic injection molding process. *IEEE Trans. Ind. Electron.*, 968–975 (2001)

12. Negnevitsky, M.: *Artificial Intelligence: A Guide to Intelligent Systems*, 2nd ed. Addison-Wesley, Harlow (2005), pp. 217–240
13. Netadelica (June 2, 2010), <http://www.netadelica.com/ga/>
14. Ogata, K.: *Modern Control Engineering*, 3rd ed. Prentice Hall, Upper Saddle River (1997), pp. 1–9 and 150–157
15. Passino, K.M., Yurkovich, S.: *Fuzzy Control*. Addison Wesley Longman, California (1998), p. 22
16. Phillips, C.L., Harbor, R.D.: *Feedback Control System*, 4th ed. Prentice Hall, Upper Saddle River (2000), pp. 101–114
17. Todd, D.: *Multiple criteria genetic algorithms in engineering design and operation*. PhD Thesis, University of Newcastle (1997)
18. Torres, G.L., Carvalho, M.A., Borges, L.E., Pinto, J.O.: Fitting fuzzy membership functions using genetic algorithms. In: 2000 IEEE International Conference on System, Man, and Cybernetics, vol. 1, pp. 387–392 (2000)
19. Tsai, C.C., Lu, C.H.: Multivariable self-tuning temperature control for plastic injection molding process. *IEEE Trans. Ind. Appl.* **34**(2), 310–318 (1998)
20. Wong, M.L., Yam, Y., Baranyi, P.: Representing membership functions as elements in function space. In: *Proceedings of the 2001 American Control Conference*, vol. 3, pp. 1922–1927 (2001)
21. Yusuf, I.: *Analysis the controller temperature set for maintained the melting point of plastic extruders by FLC*. Thesis, University of Jayabaya Indonesia (2005)
22. Yusuf, I., Iksan, N., Suryana Herman, N.: A temperature control for plastic extruder used fuzzy genetic algorithms. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International Multiconference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 1075–1080 (2010)

Automatic Weight Selection and Fixed-Structure Cascade Controller for a Quadratic Boost Converter

Somyot Kaitwanidvilai and Pitsanu Srithongchai

Abstract In this paper, a new technique for designing a robust cascade controller for a quadratic boost converter is proposed. A single performance index in H infinity loop shaping control called stability margin is adopted as the objective function in the proposed optimization control problem; GA is used to solve this problem to evaluate the optimal controller. Necessary conditions of cascade control scheme are adopted as the constraints in the optimization problem. In addition, pre-compensator weight, which is normally difficult to be selected, is simultaneously determined with the controller. Comparative study with the conventional H infinity loop shaping is presented. Finally, simulation results verify the effectiveness of the proposed technique.

Keywords H infinity loop shaping control · Genetic algorithm · Quadratic boost converter · Fixed-structure robust control

1 Introduction

Several advantages of quadratic DC-DC converter such as reduction of the resonance mode in DC-DC converter, simple circuit, etc. have been presented in the previous research works [1–4]. The design of robust controller for this converter is needed to enhance both the performance and the robustness of the controlled system. Although the standard technique such as H infinity optimal control can provide a feasible way to design the robust controller; however, the resulting controller in this approach is normally complicated with high order, making it difficult to implement in practice. In addition, weight selection in this technique, which is not an easy task, is normally carried out by trial and error method and needs the expert

S. Kaitwanidvilai (✉)

Center of Excellence for Innovative Energy Systems, King Mongkut's Institute of Technology
Ladkrabang, Bangkok 10520, Thailand
e-mail: drsomyotk@gmail.com

knowledge to find the appropriate weights. To overcome this problem, we propose an algorithm, a robust cascade controller designed by GA, to design a robust controller for a quadratic DC-DC boost converter. In the proposed technique, inverse of infinity norm from disturbances to states is formulated as the fitness function in GA. The advantages of simple structure, controller structure selectable, and robustness are achieved by the proposed technique. In addition, performance weight, which is normally difficult to be obtained, is simultaneously determined by GA. This reduces the difficulty of weight selection in the conventional robust loop shaping design.

The remainder of this paper is shown as follows. Section 2 illustrates the converter model and the conventional H infinity loop shaping technique. Section 3 describes the proposed technique. GA is also briefly described in this section. Simulations and results are shown in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Converter Model and Conventional Loop Shaping Technique

2.1 Converter Model

Figure 1 shows the typical circuit of a quadratic boost converter. The dynamic model of current and voltage loops in the cascade control scheme of this circuit can be expressed as [1, 2, 5]:

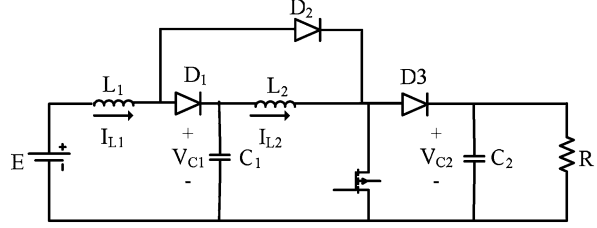
$$G_{iL} = \frac{\tilde{i}_s(s)}{\tilde{d}(s)} = K_1 \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \quad (1)$$

$$G_v = \frac{\tilde{v}(s)}{\tilde{i}_s(s)} = \frac{m_3 s^3 + m_2 s^2 + m_1 s + m_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (2)$$

where

$$\begin{aligned} K_1 &= \frac{E}{(1-U)^3 R}, & a_3 &= \frac{R(1-D)^2}{L_1} + \frac{R(1-D)}{L_2} \\ a_2 &= \frac{(1-D)^2}{L_1 C_2} + \frac{2(1-D)}{L_2 C_2} + \frac{1}{L_1 C_1} - \frac{1}{L_2 C_1 (1-D)} \\ a_1 &= \frac{2(2-D)(1-D)^2 R}{L_1 C_1 L_2} + \frac{R(1-D)^4}{L_2 C_2 L_1} + \frac{1}{L_1 C_1 C_2 R} - \frac{1}{L_2 C_1 C_2 R (1-D)} \\ a_0 &= \frac{4(1-D)^2 + 3(1-D)^3}{L_1 L_2 C_1 C_2}, & b_3 &= \frac{1}{RC_2} \\ b_2 &= \frac{1}{L_2 C_1} + \frac{(1-D)^2}{L_2 C_2} + \frac{(1-D)^2}{L_1 C_1}, & b_1 &= \frac{1}{L_2 C_2 C_1 R} + \frac{(1-D)^2}{L_1 C_2 C_1 R} \end{aligned}$$

Fig. 1 A quadratic boost converter



$$b_0 = \frac{(1-D)^4}{L_2 C_2 C_1 L_1}, \quad m_3 = -\frac{E}{RC_2(1-D)^3}, \quad m_2 = \frac{E}{L_2 C_2(1-D)}$$

$$m_1 = \frac{-E(2L_1 + L_2(1-D)^2)}{RL_1 C_1 L_2 C_2(1-D)^3}, \quad m_0 = \frac{2E(1-D)}{L_1 C_1 L_2 C_2}$$

D , d are nominal duty cycle and duty cycle, respectively; L_1 , L_2 , C_1 , C_2 , R are the component values of the converter shown in Fig. 1; E is the nominal input voltage; G_{iL} and G_v are the dynamic models of current and voltage loops; $i_s = i_{L1} + i_{L2}$.

2.2 The Conventional H_∞ Loop Shaping Control

H infinity loop shaping control was first introduced by McFarlane [6]. In this design, desired open loop shape in frequency domain is specified by shaping the open loop of the system, G , with the weighting functions, pre-compensator (W_1) and post-compensator (W_2). The shaped plant can be written as:

$$G_s = W_1 G W_2 \quad (3)$$

$$G_\Delta = (N_s + \Delta_{N_s})(M_s + \Delta_{M_s})^{-1} \quad (4)$$

where Δ_{N_s} and Δ_{M_s} are the uncertainty transfer functions in the nominator and denominator factors, respectively. G_Δ is the shaped plant with uncertainty. $\|\Delta_{N_s}, \Delta_{M_s}\|_\infty \leq \varepsilon$, where ε is the stability margin. The design steps of H infinity loop shaping can be briefly described as follows:

Step 1 Specify the pre- and post-compensator weights for achieving the desired open loop shape.

Step 2 Find the optimal stability margin (ε_{opt}) by solving the following equation.

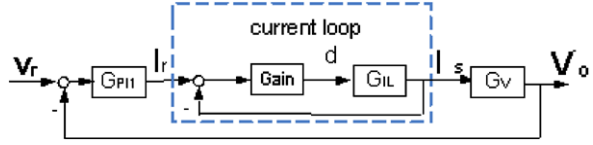
$$\gamma_{opt} = \varepsilon_{opt}^{-1} = \inf_{stabK} \left\| \begin{bmatrix} I \\ K \end{bmatrix} (I - G_s K)^{-1} M_s^{-1} \right\|_\infty \quad (5)$$

If (ε_{opt}) is too low, then go to Step 1 to select the new weights.

Step 3 Select the stability margin ($\varepsilon < \varepsilon_{opt}$) and then synthesize the controller, K_∞ , by solving the following inequality.

$$\begin{aligned} \|T_{zw}\|_\infty &= \left\| \begin{bmatrix} I \\ K_\infty \end{bmatrix} (I - G_s K_\infty)^{-1} M_s^{-1} \right\|_\infty \\ &= \left\| \begin{bmatrix} I \\ K_\infty \end{bmatrix} (I - G_s K_\infty)^{-1} [I \quad G_s] \right\|_\infty \leq \varepsilon^{-1} \end{aligned} \quad (6)$$

Fig. 2 Cascade control scheme for a quadratic boost converter



Step 4 Final controller (K) is

$$K = W_1 K_\infty W_2 \quad (7)$$

3 Fixed-Structure Robust Loop Shaping Control

Although the robust loop shaping technique is an efficient technique to design a robust controller; however, the final controller designed by this approach is usually high order and complicated. To overcome this problem, we propose a GA based fixed-structure robust loop shaping control to design a fixed-structure robust controller. The proposed technique can be described as follows.

The structure of the proposed controller is shown in Fig. 2.

In this paper, we selected PI controller as the outer loop controller and P controller as the inner loop controller. If $G_{innerloop}$ is the transfer function of the closed loop system of the current loop, thus, $G_s = W_1 G_{innerloop} G_v$ and the stability margin in (6) can be written as:

$$\left\| \left[\begin{array}{c} I \\ W_1^{-1} G_{PI1} \end{array} \right] (I - G_s W_1^{-1} G_{PI1})^{-1} [I \quad G_s] \right\|_\infty^{-1} \quad (8)$$

Consequently, our design objective is to find the optimal gains of the P controller of current loop, the PI controller (G_{PI1}), and the weight W_1 such that the stability margin in (8) is maximized. To achieve the cascade controller, a constraint, which is “the bandwidth of inner loop must be much higher than that of the outer loop” is added to the optimization problem. The following steps are the proposed design.

Step 1 Specify the structures of weight and controller. Select the post-compensator weight as I .

Step 2 The structure of the voltage loop controller is $G_{PI1}(p)$, thus, based on (7),

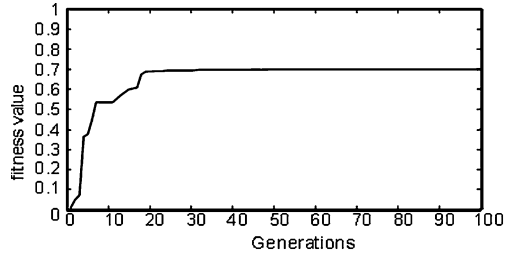
$$K_\infty = W(x)^{-1} G_{PI1}(p) \quad (9)$$

By substituting (9) into (6), the infinity norm of transfer function from disturbances to states, subjected to be minimized, can be written as:

$$\begin{aligned} J_{cost} &= \frac{1}{\varepsilon} = \gamma = \|T_{zw}\|_\infty \\ &= \left\| \left[\begin{array}{c} I \\ W(x)^{-1} G_{PI1} \end{array} \right] (I - G_s W(x)^{-1} G_{PI1}(p))^{-1} [I \quad G_s] \right\|_\infty \end{aligned} \quad (10)$$

where $G_s = W_1 G_{innerloop} G_v$.

Fig. 3 Fitness versus generations in GA optimization



Subject to BW (inner loop) $> 5BW$ (outer loop). BW is denoted as bandwidth.

Step 3 Use GA to find the optimal parameter, p^* , gain, and x^* . The followings briefly describe the GA.

In the proposed technique, GA is adopted in the control synthesis. This algorithm applies the concept of chromosomes and the genetic operations of crossover, mutation and reproduction. At each step, called generation, fitness value of each chromosome in population is evaluated by using fitness function. Chromosome, which has the maximum fitness value, is kept as a solution in the current generation. The new population of the next generation is obtained by performing the genetic operators such as crossover, mutation, and reproduction. In this paper, a roulette wheel method is used for chromosome selection. In this method, chromosome with high fitness value has high chance to be selected. Operation type selection, mutation, reproduction, or crossover depends on the pre-specified operation's probability. Normally, chromosome in genetic population is coded as binary number. However, for the real number problem, decoding binary number to floating number is applied.

4 Simulation Results

In our study, the converter parameters are given as follows: $C_1 = 22 \mu\text{F}$, $C_2 = 100 \mu\text{F}$, $L_1 = 90 \mu\text{H}$, $L_2 = 382 \mu\text{H}$, load $R = 100$ ohms. GA is adopted to find the solution of above optimization problem. Weight parameters, gain in PI controller, and gain in P controller, are set as the chromosome in GA. Constraints of time domain specifications, i.e. settling time < 0.05 sec., overshoot $< 0.5\%$ are also adopted in this optimization problem. When running GA for 49 generations, an optimal solution is obtained.

Resulting weight and controller are shown in Table 1. Conventional H infinity control with the same weight and inner loop current controller gain is adopted to design the controller for comparison purpose. The full order H infinity controller is designed as (11).

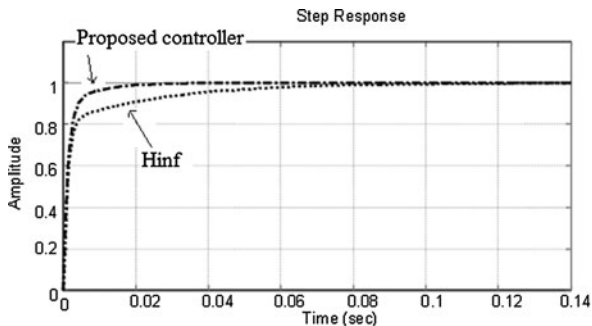
$$H_\infty = \frac{a^*c}{b} \quad (11)$$

where

$$a = \frac{(0.5665s + 18.05)}{s}$$

Table 1 Resulting controllers and their stability margins

	Proposed controller	Conventional H_∞ loop shaping
Weight	$(0.5665s + 18.05)/s$	$(0.5665s + 18.05)/s$
Controller (outer loop)	$(0.4418s + 53.97)/s$	H_∞ in (11) (9th order controller)
Controller (inner loop)	2.74	2.74
Stability margin	0.70	0.745

Fig. 4 Step response of the proposed controller and H_∞ controller

$$\begin{aligned}
 b &= s^8 + 1.084 \times 10^6 s^7 + 1.068 \times 10^{10} s^6 + 5.491 \times 10^{14} s^5 + 4.799 \times 10^{18} s^4 \\
 &\quad + 6.984 \times 10^{22} s^3 + 5.482 \times 10^{26} s^2 + 1.138 \times 10^{29} s + 3.113 \times 10^{30} \\
 c &= 6037s^7 + 6.499 \times 10^9 s^6 + 1.223 \times 10^{13} s^5 + 3.214 \times 10^{18} s^4 \\
 &\quad + 3.013 \times 10^{21} s^3 + 3.968 \times 10^{26} s^2 + 8.896 \times 10^{28} s + 3.166 \times 10^{30}
 \end{aligned}$$

As seen in this table, the stability margin of the proposed controller is almost the same as the conventional full order H infinity controller; however, the order of our controller is much lower than that of the conventional controller. This makes it easy to be implemented in practice. Since analog controller is normally used to design the converter controller, our technique is more feasible than the conventional robust control.

Time domain responses of both controllers are shown in Figs. 3 and 4. As seen in these figures, our proposed technique gains better response in terms of fast rise time and fast settling time.

5 Conclusions

In this paper, the design of high-performance and robust controller for a quadratic DC-DC converter using Genetic Algorithm has been proposed. Results show that the order of the proposed controller is much lower than that of the conventional robust loop shaping controller. In addition, by the proposed technique, performance weight which is not easy to be specified, can be simultaneously evaluated with the

controller. The tracking performance specifications can be adopted as the constraint in the proposed optimization problem.

Acknowledgements This work was supported by the DSTAR, KMITL and NECTEC, NSTDA and the King Mongkut's Institute of Technology Ladkrabang Research Fund.

References

1. Morales-Saldana, J.A., Galarza-Quirino, R., Leyva-Ramos, J., Carbajal-Gutiérrez, E.E., Ortiz-Lopez, M.G.: Multiloop controller design for a quadratic boost converter. *Electr. Power Appl., IET* **1**, 362–367 (2007)
2. Ortiz-Lopez, M.G., Leyva-Ramos, J., Carbajal-Gutierrez, E.E., Morales-Saldana, J.A.: Modelling and analysis of switch-mode cascade converters with a single active switch. *Power Electron., IET* **1**, 478–487 (2008)
3. Harb, A.M., Smadi, I.A.: Tracking control of DC motors via mimo nonlinear fuzzy control. *Chaos Solitons Fractals* **42**(2), 702–710 (30 October 2009)
4. Carbajal-Gutiérrez, E., Morales-Saldana, J.A., Leyva-Ramos, J.: Modeling of a single-switch quadratic buck converter. *IEEE Trans. Aerosp. Electron. Syst.* **41**, 1450–1456 (2005)
5. Srithongchai, P., Kaitwanidvilai, S.: Robust fixed-structure cascade controller for a quadratic boost converter. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 1030–1033 (2010)
6. McFarlane, D., Glover, K.: A loop shaping design procedure using H_∞ synthesis. *IEEE Trans. Automat. Contr.* **37**(6), (1992)

Availability Studies and Solutions for Wheeled Mobile Robots

Adrian Korodi and Toma L. Dragomir

Abstract The need of an increased availability in the field of robotics is essential. The subjects of this work are the wheeled mobile robots, and their key feature is the movement. The movement has to be realized as safely as possible. The safety implies faultless behavior, which means achieving the specification limits in any situation. A critical part in the wheeled mobile robots movement is the localization module. The first objective of the paper is to provide an availability analysis for wheeled mobile robots regarding the localization module. The first analysis focuses on a distance sensor used to detect static or dynamic obstacles from the environment and the second analysis is related to the entire localization module, the main part of the faults coming from odometry errors. The second objective of the paper is to realize a synthesis of redundant procedures meant to increase the value of the overall availability of the mobile robots.

Keywords Availability · Fault tolerance · Markov models · Odometry errors · Wheeled mobile robot

1 Introduction

A high reliability and availability level for robotic systems is an increasing demand, especially for wheeled mobile robots which are moving in completely unknown or partially known environments. The mobile robots are repairable systems, and therefore the availability is essential. The researches focused on reliability and availability analysis are in a small number [1, 3, 4, 7, 9] and many of them are incomplete. They are pointing out a reduced level of availability and a need of redundant structures in order to increase it.

A. Korodi (✉)

“Politehnica” University of Timisoara, Faculty of Automation and Computer Science,
Department of Automation and Applied Informatics, Bd. Vasile Parvan no. 2, 300223, Timisoara,
Romania

e-mail: adrian.korodi@aut.upt.ro

One critical and essential point for the mobile robots which are moving in partially known environments is the localization module. If it fails, the entire system fails, not being able to determine its own position, to avoid static obstacles, or not being able to detect the approaching mobile objects. The failures in the localization module are representing 25% of the total number of failures.

In consequence an availability analysis method based on Markov models, and two availability analyses for wheeled mobile robots will be provided in this paper. Both availability studies are focused on the localization module. The first study determines the availability for an average distance sensor (present in every wheeled mobile robot), and it points out the availability after a fault tolerant controller is implemented [5] in order to tolerate certain types of faults of the distance sensor. The second study focused on the entire localization module and it provides an availability study for a wheeled mobile robot which is moving in partially known environments. Also, being known that the majority of the localization failures are caused by odometry errors, the study determines the availability of the wheeled mobile robot after a correction module based on image processing is implemented [6], showing the results in a comparative manner.

2 First Availability Analysis

The current paragraph realizes an availability study over a usual distance sensor used in the localization module by the wheeled mobile robots. The mean time between failures (MTBF) is difficult to estimate because of the various types of sensors which may be used. Regarding the various received information regarding the MTBF for an average distance sensor, the following time values will be considered: 6 months, respectively 5 months.

In the first situation the MTBF will be 180 days and therefore the failure rate will be $\lambda_i = 0.000231 \text{ hours}^{-1}$. Regarding the fact that the paper wants to point out also how the availability can be increased, types of failures are defined and the corresponding failure rates are analyzed. The types of possible faults are [5]:

- *a sudden fault* is defined as the variation of the amplitude of the signal provided by the sensor (e) greater than a prescribed value (p_a), over consecutive sampling periods:

$$e[t] - e[t - 1] > p_a \quad (1)$$

- *an evolutive fault* is characterized by a temporized failure of the sensor, caused by certain wear out, pointed out through an amplified or attenuated signal provided by the sensor.

A permanent fault is defined as a sudden fault maintained over n_p sampling periods or as an evolutive fault which exceeds the acceptable limits.

Examining the general MTBF and following a data analysis received from various sensor producers, the author appreciates that the failure rates for evolutive,

sudden and permanent faults are $\lambda_e = 0.000124 \text{ hours}^{-1}$, $\lambda_b = 0.000077 \text{ hours}^{-1}$, respectively $\lambda_p = 0.000086 \text{ hours}^{-1}$.

The second situation corresponds to a 150 days MTBF, resulting the following general failure rate $\lambda_i = 0.000277 \text{ hours}^{-1}$. On the same bases that for the first situation, the following failure rates have resulted: $\lambda_e = 0.000123 \text{ hours}^{-1}$ for evolutive faults, $\lambda_b = 0.000079 \text{ hours}^{-1}$ for sudden faults, respectively $\lambda_p = 0.000125 \text{ hours}^{-1}$ for permanent faults.

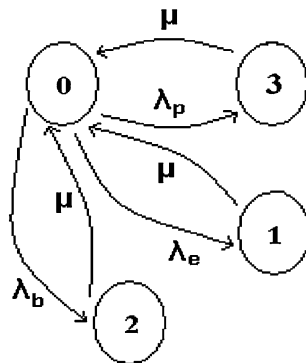
Regarding the repair rate for the sensors, following the affirmations made by big sensor producers (Schneider, Massa, etc.), the sensors must be almost always replaced after a failure, the repairing being impossible. Therefore, after a failure at the distance sensor, regardless of its type (ultrasonic or infrared), a minimum 2 weeks mean downtime (MDT) is estimated. This downtime includes the buying and the shipping of the new sensor and also the faulty sensor replacing procedure. So, the considered repair rate for the availability study will be $\mu = 0.0029 \text{ hours}^{-1}$.

In order to realize an availability study, the Markov modeling will be used. The system's states are:

- state 0 – the state of system success, the system is not affected by any type of fault at the distance sensor;
- state 1 – the state in which the distance sensor suffers an evolutive fault;
- state 2 – the state in which the distance sensor suffers a sudden fault;
- state 3 – the state in which the distance sensor is affected by a permanent fault.

The corresponding model is illustrated in Fig. 1:

Fig. 1 The established model for the availability analysis



The transitory processes associated with the probabilities of the system to be in one of the four states of the model presented in Fig. 1 can be described by the following equations:

$$\begin{aligned}
 P_0(t + \Delta t) = & P_0(t)(1 - \lambda_p \Delta t - \lambda_b \Delta t - \lambda_e \Delta t) \\
 & + P_1(t)\mu \Delta t + P_2(t)\mu \Delta t + P_3(t)\mu \Delta t
 \end{aligned}$$

$$\begin{aligned}
P_1(t + \Delta t) &= P_0(t)\lambda_e\Delta t + P_1(t)(1 - \mu\Delta t) \\
P_2(t + \Delta t) &= P_0(t)\lambda_b\Delta t + P_2(t)(1 - \mu\Delta t) \\
P_3(t + \Delta t) &= P_0(t)\lambda_p\Delta t + P_3(t)(1 - \mu\Delta t)
\end{aligned} \tag{2}$$

Realizing $\Delta t \rightarrow 0$, the state equations are obtained. These are helping in the study of the transition dynamics and availability evaluation:

$$\begin{cases} \dot{P}_0(t) = -(\lambda_e + \lambda_b + \lambda_p) \cdot P_0(t) + \mu \cdot (P_1(t) + P_2(t) + P_3(t)) \\ \dot{P}_1(t) = \lambda_e \cdot P_0(t) - \mu \cdot P_1(t) \\ \dot{P}_2(t) = \lambda_b \cdot P_0(t) - \mu \cdot P_2(t) \\ \dot{P}_3(t) = \lambda_p \cdot P_0(t) - \mu \cdot P_3(t) \end{cases} \tag{3}$$

3 First Availability Improvement

The paragraph wants to emphasize one possibility to improve the availability having the distance sensor and its possible faults described in Sect. 2. A possibility to increase the availability level in the current situation is to use a fault tolerant controller like the one it is proposed in [5]. The purpose is not to detail the referred paper and therefore the following brief description is provided: the evolutive and sudden faults can be tolerated using a fault tolerant controller. The fault tolerant controller developed in [5] is composed by: a basic interpolative controller which uses a three-dimensional table, a reference-based redundancy to detect the faults and a correction module in order to provide the correct input for the basic interpolative controller if the distance sensor suffers an evolutive fault.

The paragraph will do a comparative availability analysis, for the situation presented in the previous paragraph and the situation in which a fault tolerant controller is used to tolerate evolutive and sudden faults. There will be two case studies regarding the distance sensor data regarding from Sect. 2.

Figure 3 illustrates in a comparative manner the availability evolutions. A_{initial} represents the initial systems availability (without tolerating sudden and evolutive faults), and A_{final} , represents the availability of the improved system.

In order to calculate A_{initial} , state ① was considered the only success state (because the evolutive and sudden faults are not tolerated by the initial system). When calculating A_{final} availability, the success states of the system were ①, ② and ③, because the system tolerates evolutive and sudden faults of the distance sensor.

The Simulink scheme necessary for the study, corresponding to the formula (3), is illustrated in Fig. 2.

As it can be seen in Fig. 3, the availability is improving by 6.5% when the fault tolerant controller is used.

For the second situation (MTBF = 150 days), the concept is the same and after the simulations, the following evolutions (Fig. 4) are obtained for the availabilities A_{initial} and A_{final} .

As it can be seen in the figure, the availability is improving with 6–6.5% when the fault tolerant controller is used.

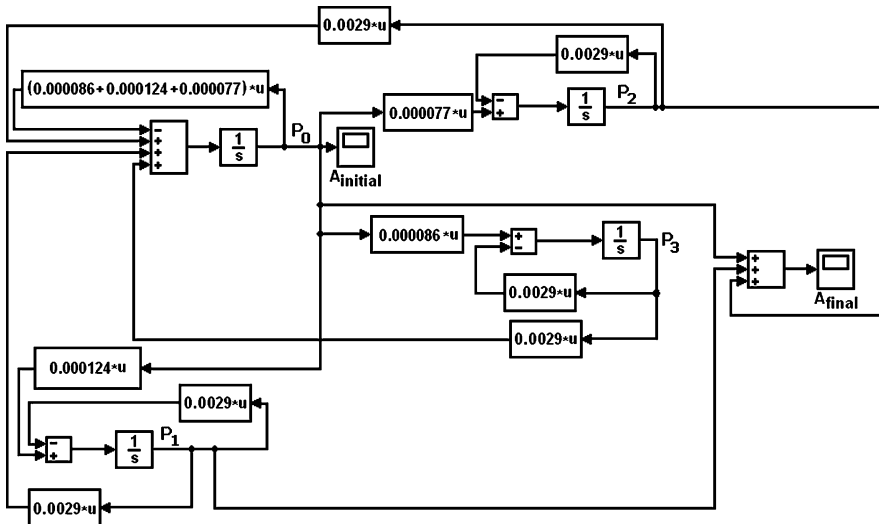
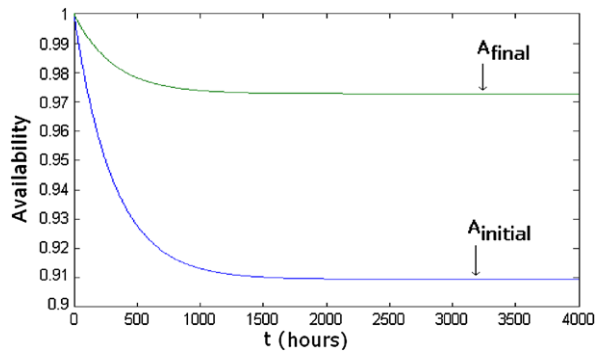


Fig. 2 The Simulink scheme associated with the availability analysis

Fig. 3 The availability evolutions for the first situation study (MTBF = 180 days)



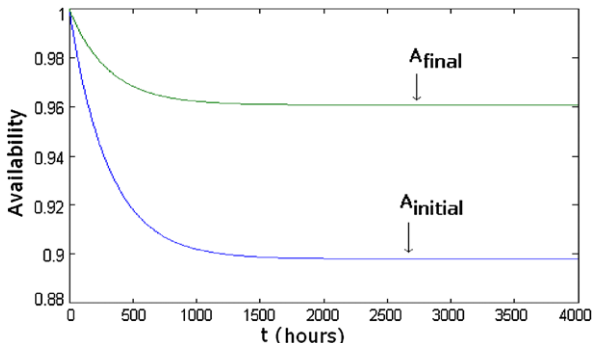
4 Second Availability Analysis

The following two paragraphs are realizing an availability and reliability study regarding the influence of localization error correction during the movement of a wheeled mobile robot.

The literature provides an extremely small number of availability studies for wheeled mobile robots which are moving in partially known environments. The ones useful for the research within this paragraph were only [1, 4, 9]. In the mentioned references, various data was presented regarding the reliability and availability of the wheeled mobile robots which are moving in partially known environments. The specific data in order to realize an availability study for different types of robots can be obtained only experimentally, after a long enough monitoring process.

As being mentioned in [1, 4, 9], the MTBF indicator for wheeled mobile robots which are moving in partially known environments is: 7 hours (from the research

Fig. 4 The availability evolutions for the second situation study (MTBF = 150 days)



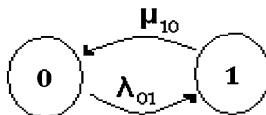
[9]) and 19.5 hours (from the research [3]). From the study realized in [1] a MTBF of 17.5 hours was determined (the robot is functioning over a period of 140 hours and within this period 8 faults were occurring).

To be able to realize the current study, it was necessary to determine the percentage of the total failures which are caused by an erroneous localization. From [9] and [1], the following conclusion can be extracted: approximately 25% of the total failures are localization failures which can be corrected using the method presented in the next paragraph.

The information until this point allows accomplishing a reliability study. But, the mobile robot is a reparable system and therefore a complete study must present an availability analysis. To realize an availability study, the MDT indicator must be known. In [4], the MDT indicator is provided, being $MDT = 60.5$ hours. The studies from [3, 4] are covering only the analysis of the inherent availability, although the operational availability is the one that is relevant. From the inherent availability study the value of the MTTR indicator is extracted, $MTTR = 1.3$ hours.

The reliability and availability study is based on Markov models. The number of states considered for a minimal analysis in this case is two: state 0, the state of success, and state 1, the failure state. The model from Fig. 5 is considered:

Fig. 5 The established model for the reliability and the availability study



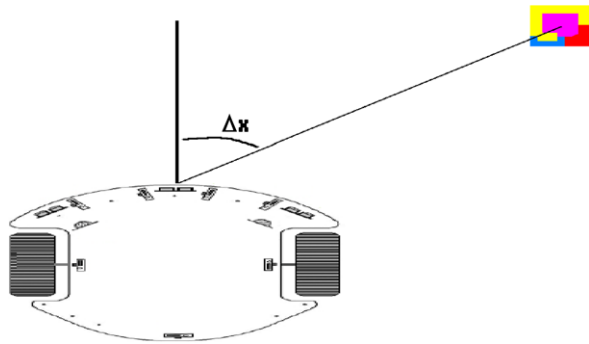
The transitory processes associated with the probabilities of being in one of the two states represented in Fig. 5 can be described by the following equations, which are taking into account the corresponding failure and repair rates:

$$\begin{aligned}
 P_0(t + \Delta t) &= P_0(t)(1 - \lambda_{01} \Delta t) + P_1(t)\mu_{10}\Delta t \\
 P_1(t + \Delta t) &= P_0(t)\lambda_{01}\Delta t + P_1(t)(1 - \mu_{10}\Delta t)
 \end{aligned}
 \tag{4}$$

Realizing $\Delta t \rightarrow 0$, the state equations are obtained. These are helping in the study of the transition dynamics, reliability and availability evaluation:

$$\begin{cases} \dot{P}_0(t) = -\lambda_{01} P_0(t) + \mu_{10} P_1(t) \\ \dot{P}_1(t) = \lambda_{01} P_0(t) - \mu_{10} P_1(t) \end{cases}
 \tag{5}$$

Fig. 6 Modifying the orientation of the wheeled mobile robot



5 Second Availability Improvement

In the previous paragraph, the percentage of localization failures from the total number of failures was found out to be significant. In the case of the wheeled mobile robots, a big number of localization failures are caused by odometry errors. The systematic odometry error corrections are studied in [2, 8], realizing a priori calibration. There is a necessity to correct also the nonsystematic odometry errors and obviously, the correction has to be realized online. An online correction module based on image processing is proposed in [6]. This method is suitable for wheeled mobile robots that are moving in dynamic environments. Using a camera placed on the mobile robot, images are taken over from visual reference (target) points placed priori in the environment. If a fault that can be caused by odometry errors is detected, the orientation of the mobile robot towards the next reference point is corrected as in Fig. 6 (identifying the visual reference point through image processing), and the robot will be able to accomplish its task.

Implementing the module described in [6], the localization for the wheeled mobile robots which are moving in partially known environments is improved and the overall availability and reliability of the system increases. A comparative availability and reliability study is realized in order to show the efficiency of using a redundant method in the localization module.

Four experiments are realized: one reliability and inherent availability study and two operational availability studies.

For the experiments, the transitions provided by the formulas (5) and (6) are implemented in Simulink, like shown in Fig. 7.

The first experiment realizes a minimal wheeled mobile robot reliability analysis using the data from [3], where the MTTF is 19.5 hours (in the [9] study the situation is worst having a $MTTF = 7$ hours). Off course that in order to realize a reliability analysis, the repair rate from (5) is considered zero. Practically, the equation which guides the study is:

$$\dot{P}_0(t) = -\lambda_{01} P_0(t) \tag{6}$$

Figure 8 shows the comparative reliability evolutions for the two situations: the basic system and the system where the redundant orientation correction method based on image processing is implemented.

Fig. 7 The Simulink scheme used to analyze the availability and the reliability of the system

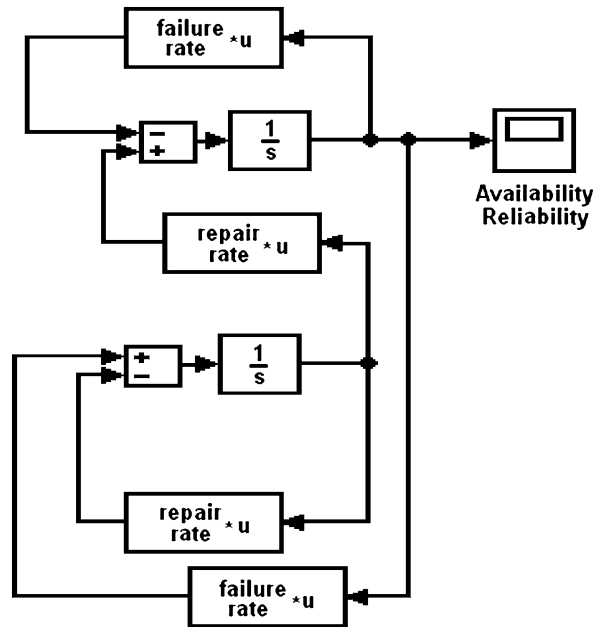
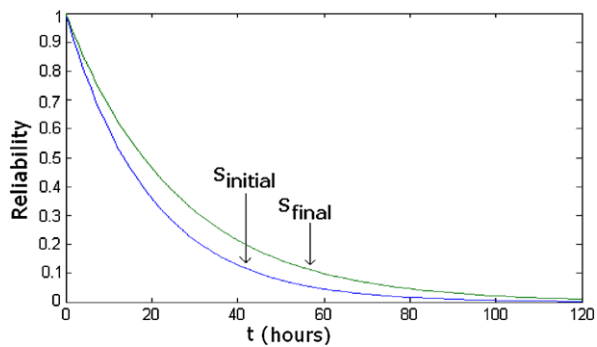


Fig. 8 The reliability evolutions for the two situations



The notations are the followings:

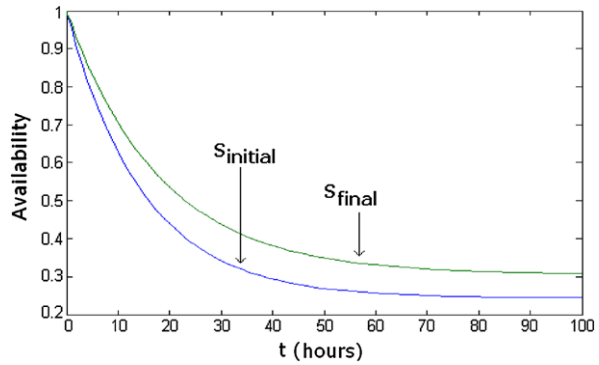
- the analyze referring to the initial system – $S_{initial}$
- the analyze referring to the system foreseen with redundant module – S_{final}

Like it was mentioned before, the MTTF is 19.5 hours, and the failure rates have the following values: $\lambda_{01} = 0.05128 \text{ hours}^{-1}$ for $S_{initial}$, and $\lambda_{01} = 0.03846 \text{ hours}^{-1}$ for S_{final} .

The following two experiments have as objective, the comparative study of the operational availability of the wheeled mobile robot for the two implemented situations, $S_{initial}$ and S_{final} .

The first experiment shows an availability analysis where the MTBF is 19.5 hours for $S_{initial}$, and consequently its failure rate is $\lambda_{01} = 0.05128 \text{ hours}^{-1}$. In this

Fig. 9 The evolution of the availabilities for the current situation

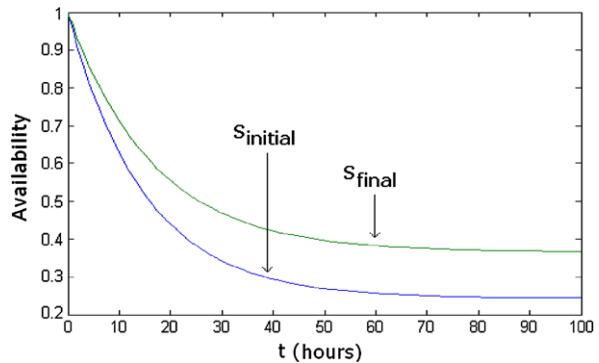


study, the objective is the operational availability which is the most relevant, and therefore the MDT indicator is 60.5 hours (as being presented in [3]), respectively the repair rate is $\mu_{10} = 0.01652 \text{ hours}^{-1}$. For S_{final} , the failure rate will be $\lambda_{01} = 0.03846 \text{ hours}^{-1}$, and the repair rate in this case is kept constant although certainly it will grow by introducing the redundant module. Figure 9 presents the comparative evolution of the availabilities, using (5).

As being observed in Fig. 9, the consequence of implementing the redundant module to correct the orientation of the mobile robot through image processing is that the overall availability is increased by approximately 7%.

The second experiment follows the same hypothesizes as the previous one, but considering the fact that the repair rate will be influenced also by the decrease of the MTBF indicator. The influence is obvious, but the percentage of which the MDT indicator is modifying is just presumed. The analysis will consider that the MDT will decrease to 45 hours and therefore the repair rate will became $\mu_{10} = 0.02222 \text{ hours}^{-1}$. Figure 10 illustrates the results of this analysis:

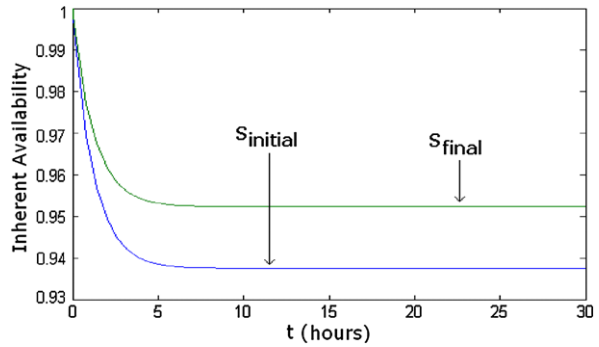
Fig. 10 The operational availability evolutions for the current situation



From Fig. 10 it can be noticed that the steady-state availability of S_{final} is bigger than the steady-state availability of S_{initial} by approximately 13%.

The last experiment is focused on the inherent availability, when the MTTF indicator is 1.3 hours (value provided in [3]). The repair rate in these conditions is

Fig. 11 The inherent availability evolution for the current situation



$\mu_{10} = 0.76923 \text{ hours}^{-1}$. The values of the failure rates are considered to be the same with the ones from the previous experiment. It is considered also that the repair rate will not be modified following the implementation of the redundant module which will correct the orientation of the wheeled mobile robot (this fact implies a minimal influence over the availability, although, certainly the repair rate will also be influenced by implementing the redundant module). Figure 11 illustrates the availability evolution for the S_{initial} system and the S_{final} system, and an inherent availability increase of approximately 1.5% of the S_{final} over the S_{initial} system can be noticed.

6 Conclusions

The paper showed two availability studies regarding the localization module of the wheeled mobile robots which are moving in partially known environments. The method illustrated and used to analyze the availability was based on Markov models.

The first study was showing the availability evolution before and after implementing a fault tolerant controller in order to tolerate certain faults of the distance sensor. The second study provided the comparative availability evolution for the wheeled mobile robot before and after implementing a redundant method based on image processing designed to correct odometry errors.

In conclusion, the necessity to analyze and increase the availability of the wheeled mobile robots is obvious and it is a permanent requirement in order to achieve a higher degree of autonomy.

References

1. Austin, D., Kouzoubov, K.: Robust, long term navigation of a mobile robot. In: Proc. IARP/IEE-RAS Joint Workshop on Technical Challenges for Dependable Robots in Human Environments (2002)
2. Borenstein, J., Feng, L.: UMBmark: a benchmark test for measuring odometry errors in mobile robots. In: Proceedings of the SPIE Conference on Mobile Robots, Philadelphia, USA, 22–26 October 1995

3. Carlson, J., Murphy, R.R.: Reliability analysis of mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003), pp. 274–281 (2003)
4. Carlson, J., Murphy, R., Nelson, A.: Follow-up analysis of mobile robot failures. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004), New Orleans, USA (2004)
5. Korodi, A.: Interpolative fault tolerant controller for a mobile robot. In: Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics MMAR, Miedzyzdroje, Poland, 28–31 August, pp. 651–656 (2006)
6. Korodi, A., Dragomir, T.L.: Correcting odometry errors for mobile robots using image processing. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 1040–1045 (2010)
7. Long, M., Murphy, R., Parker, L.: Distributed multi-agent diagnosis and recovery from sensor failures. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 2506–2513 (2003)
8. Stachera, K., Schumacher, W., Herbst, O.: Automatic identification of odometry parameters of a two-wheel mobile robot using ultrasonic sensors. In: Proceedings of the 12th IEEE International Conference on Methods and Models in Automation and Robotics MMAR, Miedzyzdroje, Poland, 28–31 August 2006
9. Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S., Siegwart, R.: Design and system integration for the expo. 02 robot. In: Proceedings of the IEEE/RSJ IROS 2002 Workshop Robots in Exhibitions, pp. 67–72 (2002)

The Use of Higher-Order Spectrum for Fault Quantification of Industrial Electric Motors

Jugrapong Treetrong

Abstract This chapter proposes a new method of electric motor fault quantification. Higher Order Spectrum (HOS) is a signal processing used as a fault quantification technique. Previous researches have shown that the faults in the stator or rotor generally show sideband frequencies around the main frequency (50 Hz) and its higher harmonics in the spectrum of the Motor Current Signature Analysis (MCSA). However in the present experimental studies such observations are not seen, but the faults in the stator or the rotor may distort the sinusoidal response of the motor RPM and the main frequency. Hence this research proposes the HOS here, namely the Bispectrum of the MCSA, because it relates to both amplitude and phase of number of harmonics in a signal. The Bispectrum with the unwrapped phase angle along its frequency is also analyzed. The tests can show that the proposed method can detect the faults accurately. The proposed method can also show that the severity level of the faults can be measured by observing the change in the heights of the Bispectrum amplitude.

Keywords Induction motors · Higher Order Spectrum (HOS) · Bispectrum · Condition monitoring · Fault detection

1 Introduction

Induction motors are the most widely used motors among different electric motors because of their high level of reliability, efficiency and safety. However, these motors are often exposed to hostile environments during operation which leads to early deterioration leading to the motor failure. It has also been observed that 30–40% of

J. Treetrong (✉)

Department of Teacher Training in Mechanical Engineering, Faculty of Technical Education,
King Mongkut's University of Technology, North Bangkok, Pibul-Songklarm, Bangkok, 10800,
Thailand

e-mail: jugrapong@yahoo.com

all recorded faults are generally related to the stator or armature faults caused due to the shorting of stator phase winding and 5–10% fault related to the rotor (broken bar and/or end ring fault) as shown by Nandi et al. [16]. Hence the condition and monitoring technique has generally been used to detect the fault at the early stage so that the remedial action can be done in much planned way to reduce the machine downtime and maintain the overall plant safety.

Motor Current Signature Analysis (MCSA) is one of the most spread procedures for health monitoring of the motor since decades. One of the main reasons for using this method is that the other methods require invasive access to the motor and they also need extra equipment/sensors for measuring the required signals. The research has been progressed in mainly two directions using the stator phase current and voltage signals – the detection of faults and the quantification of the faults by the motor parameters estimation [3, 4, 12, 13, 19]. First one is important for the quick health assessment on routine basis, however the later one useful to know the extent of the faults so that remedial action can be done quickly. There are number of the research studies that have used the spectrum of the stator phase current signal for the stator motor faults [1, 5, 10, 17, 18] and the rotor faults [2, 6, 8, 11, 14], often based on the presence of the side band frequency (related to the slip frequency) and its harmonics around the power supply frequency or/and its harmonics. However in the present experiments, the side bands are not clearly seen for both rotor and stator faults in their spectra with frequency resolution of 1.25 Hz when using the motor stator phase current signals, hence the use of the Higher Order Spectrum (HOS) [7, 9] is planned to apply for the stator phase current signal instead of the spectrum. Because the faults in the motor are expected to generate harmonic components of the motor RPM and the mains frequency in the motor current signal, hence the relation between different harmonic components in the signal is exploited using the HOS, namely the Bispectrum. The propose method is verified by testing on different motor fault conditions. The tests show that the proposed method can be useful in detection and quantification of the rotor and stator faults. This chapter is as a revised work from the corresponding conference paper that the reference can be seen from Treetrong [20]. The following sections, the concept of the Bispectrum is discussed and the Bispectrum results of the experimental cases of the induction motor with the healthy, stator winding short-circuits (stator fault) and broken rotor bars (rotor fault) conditions are presented.

2 Higher Order Spectrum (HOS)

The conventional power spectrum density (PSD) provides information on the second-order properties (i.e., energy) of a signal whereas the bi-spectrum can provide information on the signal's third-order properties. In a physical sense, the bi-spectrum provides insight into non-linear coupling between frequencies (as it involves both amplitudes and phases) of a signal compared to the traditional PSD that gives only the content of different frequencies and their amplitudes in a signal. The

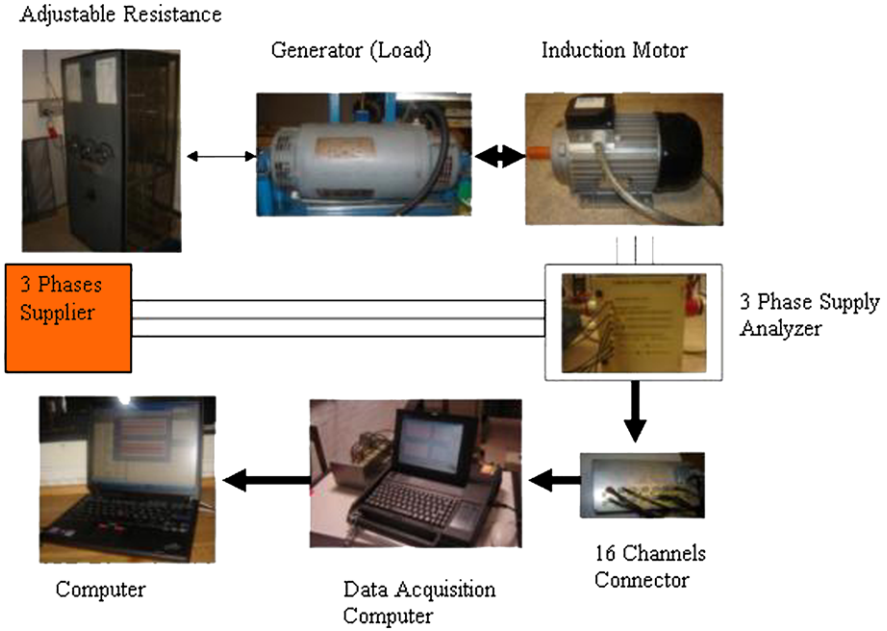
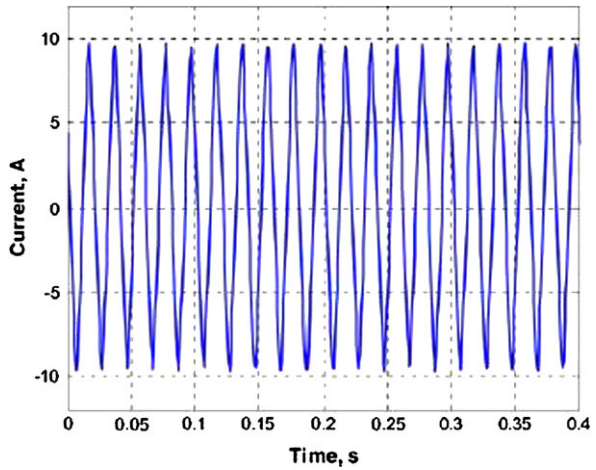


Fig. 1 Schematic of the test rig

Fig. 2 A typical current plot

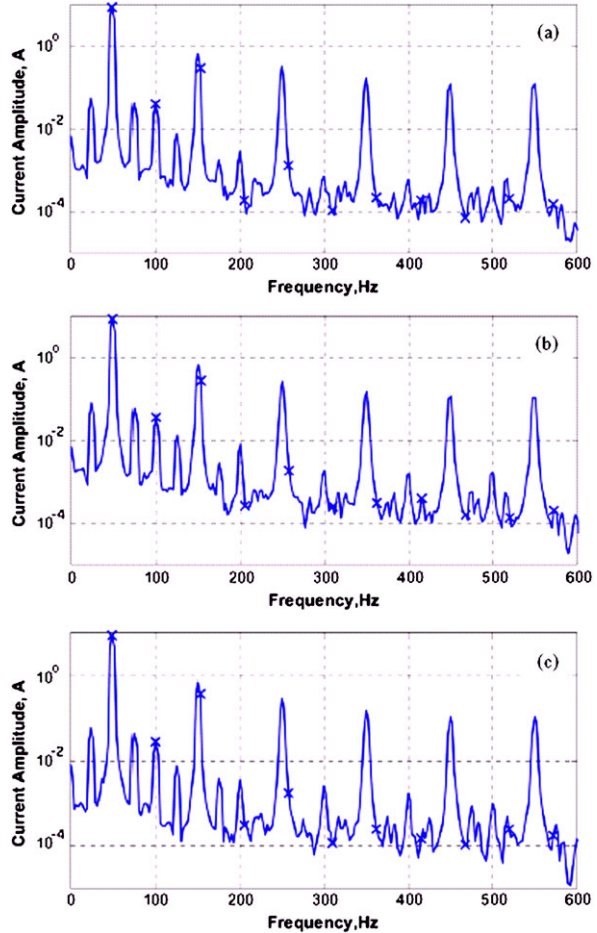


power spectrum of a time series $x(t)$ is computed by the discrete Fourier transform (DFT) of the signal as

$$\text{PSD}, \quad S_{xx}(f_k) = E[X(f_k)X^*(f_k)], \quad k = 1, 2, 3, \dots, N \quad (1)$$

where $S_{xx}(f_k)$ is the PSD, $X(f_k)$ and $X^*(f_k)$ are the DFT and its conjugate at frequency f_k for the time series $x(t)$. N is frequency points. $E[.]$ denotes the mean

Fig. 3 The spectrum plots of stator phase current:
(a) healthy motor,
(b) 15 turn's short circuit,
(c) broken rotor bars



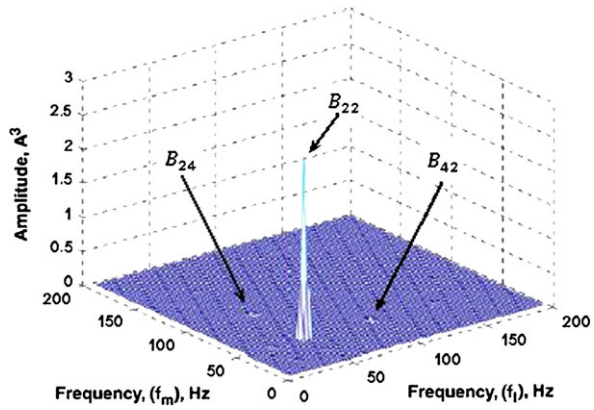
operator here. Let us assume that the time domain signal, $x(t)$, of the time length equals to t . This time signal is divided into n number of segments with some overlap and each segment contains $2N$ number of data points with sampling frequency, f_s Hz. If $X_r(f_k)$ is the FT of the r th segment, $x_r(t)$, at the frequency, f_k , then the averaged or mean PSD [7, 9] can be computed as

$$\text{Bispectrum, } B_{xxx}(f_l, f_m) = E[X(f_l)X(f_m)X^*(f_l + f_m)] \quad (2)$$

$$\mathbf{B}(f_l, f_m) = \frac{\sum_{r=1}^n \mathbf{X}_r(f_l)\mathbf{X}_r(f_m)\mathbf{X}_r^*(f_l + f_m)}{n}, \quad l + m \leq N \quad (3)$$

The Bispectrum is complex and interpreted as measuring the amount of coupling between the frequencies at f_l , f_m , and $f_l + f_m$ and is described by 'quadratic phase coupling'. It is assumed that if the frequencies, f_l and f_m are the p th and q th har-

Fig. 4 The Bispectrum plot of the stator phase current for the healthy motor



monics of the motor RPM then the component of the Bispectrum, $B(f_l, f_m)$ can represent as B_{pq} for better understanding.

3 Experimental Study

The schematic of the test rig is shown in Fig. 1. The test rig consists of an induction motor (4 kW, 1400 RPM) with load cell with a facility to collect the 3-phase current data directly to the PC at the user define sampling frequency.

The experiments are tested on these 3 different conditions – Healthy, Stator Fault and Rotor Faults at different load conditions. The data are collected at the sampling frequency of 1280 samples/s. The motor of the stator fault type can be adjusted into 3 differences of the stator short circuits – 5 turn short circuit, 10 turn short circuit and 15 turn short circuit and the motor of the rotor fault type is broken rotor bars.

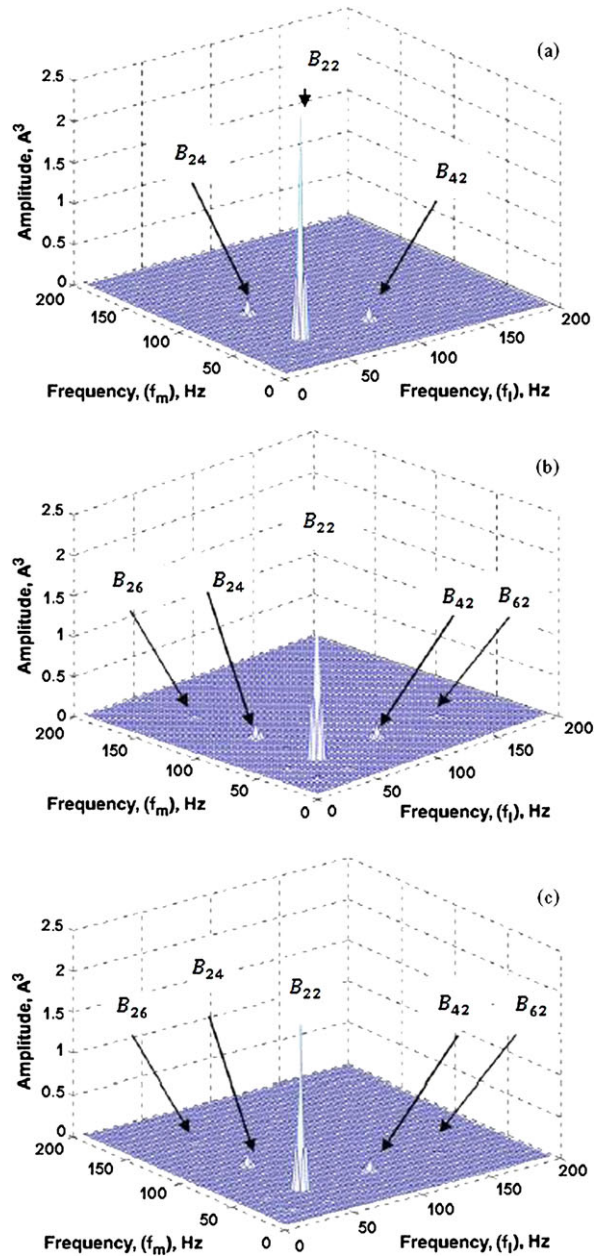
4 Result Analysis

A typical current plot for the healthy motor operating at 100% load is shown in Fig. 2. The rated current for the motor is close to 10 Amperes.

4.1 Current Spectrum

The amplitude spectrums are firstly applied for processing the stator current signals from all the experimental data of each motor condition. The frequency resolution is kept 1.25 Hz with 90% overlap and number of average 82 for all the signal processing.

Fig. 5 The Bispectrums plots of the stator phase current:
(a) 5 turns short circuit,
(b) 10 turns short circuit, and
(c) 15 turns short circuit



The computation time using the Pentium-IV PC for both the spectrum and Bispectrum is less than 30 s which it can see that the processing is definitely quick for the health monitoring purpose. Few typical plots for the amplitude spectrum of different conditions at full load are shown in Fig. 3. The amplitude spectrum is difficult

Fig. 6 The Bispectrum plot of the stator phase current for the broken rotor bar motor

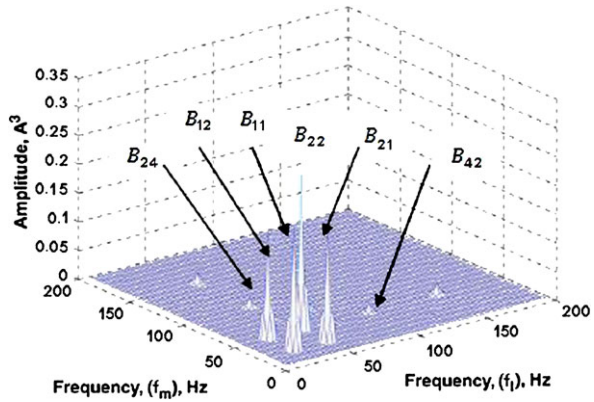


Table 1 Comparison of the Bispectrum component amplitudes for different fault conditions for the motor when operating at 100% load (unit: cubic Ampere (A^3))

	B_{11}	B_{22}	B_{12} (B_{21})	B_{24} (B_{24})	B_{26} (B_{62})
Healthy	0.025	2.70	0.020	0.080	0.018
Faulty Rotor	0.220	0.28	0.190	0.018	0.023
Faulty Stator (5 turn)	0.024	2.60	0.018	0.22	0.015
Faulty Stator (10 turn)	0.050	2.30	0.039	0.19	0.042
Faulty Stator (15 turn)	0.030	2.00	0.023	0.17	0.022

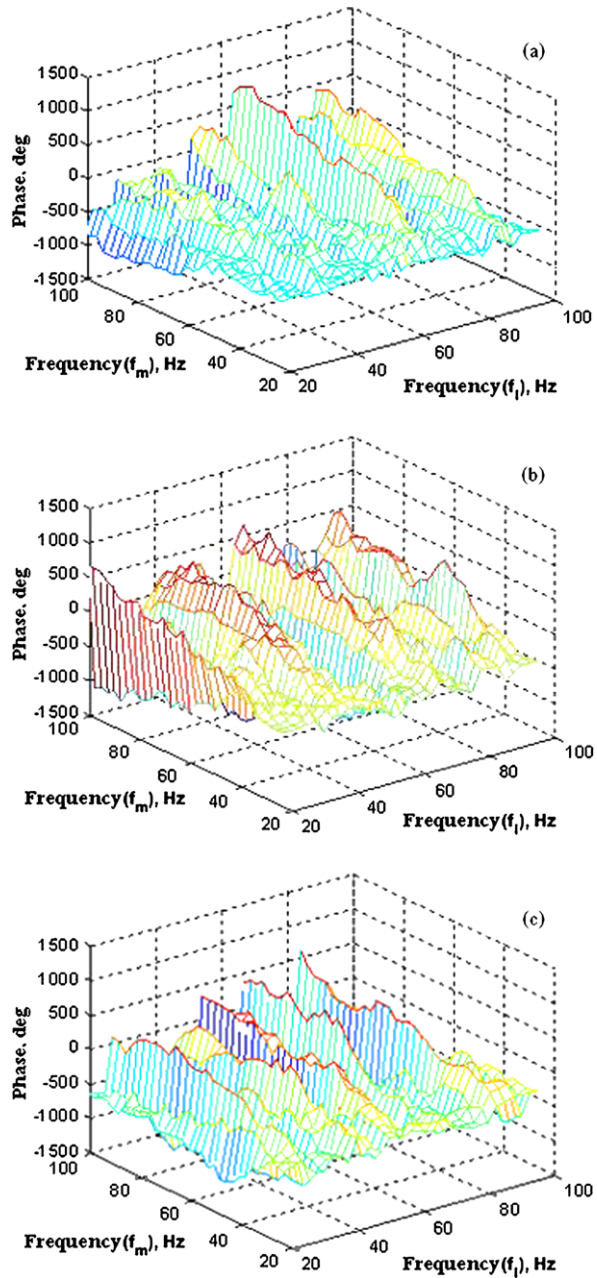
to identify the faults with the frequency resolution of 1.25 Hz of the current signals. All spectrums almost look identical. In all cases, machine RPM (1x component) and its higher harmonics (2x, 3x, ...) are present, but no side band frequency at the main frequency related to the slip frequency is seen to identify the fault as suggested in the earlier studies. Thus, it can be concluded based on the experiments that the spectrum is not showing any peaks other than machine RPM which indicates the faults by the relation between frequency and the slip frequency of the current signal. Thus, the identification based on side-band is not possible for the present experimental cases.

4.2 Current Bispectrum

The Bispectrum of stator phase currents from the 3 different motor conditions seem to identify the faults as seen from Figs. 4, 5, 6.

The peaks in the Bispectrum plots are indicated by B_{11} , B_{12} , B_{21} and so on. Here B_{11} means the relation of 1x, 1x and 2x components, B_{12} ($= B_{21}$) the relation of 1x, 2x and 3x ($1x + 2x$) components in a signal. In the Bispectrum, the only significant peak B_{22} can be seen for the healthy motor condition. However, for the faulty stator cases the amplitude of the Bispectrum component of the peak B_{22} decreases when

Fig. 7 The unwrapped phase plots of the Bispectrum, (a) healthy, (b) stator fault – 5 turns short circuit, (c) rotor fault – broken bars



the level of the stator faults increases and the significant increase in the amplitude of the Bispectrum component of the peak B_{24} ($= B_{42}$), compared to the healthy condition, nearly 2.4 times. Other peaks, B_{11} , B_{12} ($= B_{21}$), and B_{26} ($= B_{62}$) observed to

remain comparable with the amplitudes for the healthy condition (as seen in Figs. 4 and 5). However, in the case of the faulty rotor case the peak B_{22} decreases significantly compared to the healthy condition and the faulty stator cases (nearly 0.10 times) (as seen in Fig. 6), but other peaks, B_{11} and B_{12} ($= B_{21}$) increase significantly (nearly 8–10 times) compared to the healthy and the stator fault cases. These observations are also summarized in Table 1.

Hence based on the observation, it can be concluded that the Bispectrum of the phase current signal can identify and quantify the severity level of the stator and the rotor faults.

The robustness of the proposed method is also confirmed by testing on different load levels of the motors at no load, 25% load, 50% load and 75% load for the healthy, the stator fault and the rotor fault cases. The tests can show that the Bispectrum is consistent with the different load conditions. Additionally, the motors are also tested by the Bispectrum with the unwrapped phase angle, The typical phase plots of the method along the frequency, f_m and f_l for different conditions at 100% load condition as shown in Fig. 7. Based on observation, the Bispectrum with the unwrapped phase angle can differentiate the motor faults. Different features in each condition case also indicate the potential use of the phase information in diagnosis.

5 Conclusions

The HOS, namely the Bispectrum is proposed as a new technique for motor fault quantification. The proposed method can show the relation between amplitude and phase of number of harmonics in a signal. The Bispectrum of the motor phase current can successfully be able to detect the stator and the rotor fault and also able to distinguish the stator and the rotor fault. Additionally, the method can also quantify the severity level of the faults. Thus, the proposed method can be considered to be a useful tool for motor fault detection and quantification. Moreover, the Bispectrum with the unwrapped phase angle along its frequency also shows a potential technique in condition differentiation. The future research is also planned to test the Trispectrum, another kind of the HOS.

References

1. Aroquiadassou, G., Henao, H., Capolino, G.-A.: Experimental analysis of the dq0 stator current component spectra of a 42V fault-tolerant six-phase induction machine drive with opened stator phases. In: IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, SDEMPED 2007, 6–8 Sept. 2007, pp. 52–57 (2007)
2. Ayhan, B., Chow, M.Y., Song, M.H.: Multiple signature processing-based fault detection schemes for broken rotor bar in induction motors. IEEE Trans. Energy Conv. **20**(2), 336–343 (2005)
3. Bachir, S., Tnani, S., Trigeassou, J.-C., Champenois, G.: Diagnosis by parameter estimation of stator and rotor faults occurring in induction machines. IEEE Trans. Indust. Electron. **53**(3), 963–973 (2006)

4. Barut, M., Bogosyan, S., Gokasan, M.: Speed-sensorless estimation for induction motors using extended Kalman filters. *IEEE Trans. Indust. Electron.* **54**(1), 272–280 (2007)
5. Bellini, A., Filippetti, F., Franceschini, G., Tassoni, C.: Closed-loop control impact on the diagnosis of induction motors faults. *IEEE Trans. Indust. Appl.* **36**(5), 1318–1329 (2000)
6. Bellini, A., Filippetti, F., Franceschini, G., Tassoni, C., Kliman, G.B.: Quantitative evaluation of induction motor broken bars by means of electrical signature analysis. *IEEE Trans. Indust. Appl.* **37**(5), 1248–1255 (2001)
7. Collis, W.B., White, P.R., Hammond, J.K.: Higher order spectra: the bispectrum and trispectrum. *Mech. Syst. Signal Process.* **12**(3), 375–395 (1998)
8. Didier, G., Ternisien, E., Caspary, O., Razik, H.: A new approach to detect broken rotor bars in induction machines by current spectrum analysis. *Mech. Syst. Signal Process.* **21**, 1127–1142 (2007)
9. Fackrel, J.W.A., White, P.R., Hammond, J.K., Pinnington, R.J., Parsons, T.A.: The interpretation of the bispectra of vibration signals – I. *Theory. Mech. Syst. Signal Process.* **9**(3), 257–266 (1995)
10. Henao, H., Martis, C., Capolino, G.-A.: An equivalent internal circuit of the induction machine for advanced spectral analysis. *IEEE Trans. Indust. Appl.* **40**(3), 726–734 (2004)
11. Henao, H., Razik, H., Capolino, G.-A.: Analytical approach of the stator current frequency harmonics computation for detection of induction machine rotor faults. *IEEE Trans. Indust. Appl.* **41**(3), 801–807 (2005)
12. Huang, K.S., Kent, W., Wu, Q.H., Turner, D.R.: Parameter identification of an induction machine using genetic algorithms. In: *Proceeding of the IEEE International Symposium on Computer Aid Control System Design*, Hawaii, USA (1999)
13. Huang, K.S., Kent, W., Wu, Q.H., Turner, D.R.: Effective identification of induction motor parameters based on fewer measurements. *IEEE Trans. Energy Conv.* **17**(1), 55–60 (2002)
14. Kia, S.H., Henao, H., Capolino, G.-A.: A high-resolution frequency estimation method for three-phase induction machine fault detection. *IEEE Trans. Indust. Electron.* **54**(4), 2305–2314 (2007)
15. Marques Cardoso, A.J., Cruz, S.M.A., Fonseca, D.S.B.: Inter-turn stator winding fault diagnosis in three-phase induction motors’ by Park’s vector approach. *IEEE Trans. Energy Conv.* **14**(3), 595–598 (1999)
16. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE Trans. Energy Conv.* **20**(4), 719–729 (2005)
17. Nangsue, P., Pillay, P., Conry, S.E.: Evolutionary algorithm for industrial motor parameter determination. *IEEE Trans. Energy Conv.* **14**(3), 447–453 (1999)
18. Tallam, R.M., Habetler, T.G., Harley, R.G.: Stator winding turn-fault detection for closed-loop induction motor drives. *IEEE Trans. Indust. Appl.* **39**(3), 720–724 (2003)
19. Treerong, J., Sinha, J.K., Gu, F., Ball, A.D.: A novel model parameter estimation of an induction motor using genetic algorithm. In: *Proceeding of the 10th 2008 IASTED Signal and Image Conference, SIP 2008, Kailua-Kona, HI, USA, 18–20 August 2008*, pp. 623–715 (2008)
20. Treerong, J.: Fault detection and diagnosis of induction motors based on higher-order spectrum. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 1092–1096 (2010)

A Newly Cooperative PSO – Multiple Particle Swarm Optimizers with Diverisive Curiosity, MPSO α /DC

Hong Zhang

Abstract In this paper we propose a newly multiple particle swarm optimizers with diverisive curiosity (MPSO α /DC) for enhancing the search performance. It has three outstanding features: (1) Implementing plural particle swarms in parallel to explore; (2) Finding the most suitable solution in a small limited space by a localized random search for correcting the solution found by each particle swarm; (3) Introducing diverisive curiosity into the multi-swarm to alleviate stagnation. To demonstrate the proposal's effectiveness, computer experiments on a suite of benchmark problems are carried out. We investigate its intrinsic characteristics, and compare the search performance with other methods. The obtained results show that the search performance of the MPSO α /DC is superior to that by the PSO/DC, EPSO, OPSO, and RGA/E for the given benchmark problems.

Keywords Cooperative particle swarm optimization · Hybrid computation · Localized random search · Exploitation and exploration · Diverisive and specific curiosity · Swarm intelligence

1 Introduction

As a new member of genetic and evolutionary computation, particle swarm optimization (PSO) [8, 13] has been widely applied in different fields of science, technology, and applications for solving various optimization problems [20]. This is because of that it has the plain advantages: intuitive understandability, ease of implementation, and distinct expressivity. Compared to genetic algorithms and evolu-

This paper was originally presented at IMECS 2010 [26]. This is a substantially extended version.

H. Zhang (✉)

Department of Brain Science and Engineering, Graduate School of Life Science & Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu 808-0196, Japan

e-mail: zhang@brain.kyutech.ac.jp

tionary programming [11], information exchange, intrinsic memory, and directional search are the strong points of PSO.

So far, many variants of PSO such as a particle swarm optimizer with inertia weight [21], a canonical particle swarm optimizer [4, 5], and fully informed particle swarm [14] etc. were proposed for improving the convergence and search efficiency of a particle swarm optimizer (the PSO). The principal objective of these optimizers was put in methodology, i.e. enforcing the search strategy and transmission of information in the interior of a particle swarm.

In recent years, many studies and investigations on cooperative PSO in relation to symbiosis, group behavior, and sensational synergy are in the researcher's spotlight. Consequently, there are different forms of cooperative PSO, for example, hybrid PSO, and multi-population PSO etc. were published [1, 18] with deepening on group searching. In contrast to those methods that only operate a singular swarm, various attempts to the multi-swarm approach mainly focus on the rationality of information propagation within plural swarms. Some reports showed that the results of cooperative PSO have better adaptability in optimization than uncooperative PSO [3, 12, 18].

Due to great requests to search performance and swarm intelligence, utilizing the techniques of group searching and parallel processing has become one of extremely important approaches. For promoting the development of cooperative PSO research, we propose a newly cooperative PSO – multiple particle swarm optimizers with diversive curiosity (MPSO α /DC). Comparison with the convenient cooperative PSO, it has the following outstanding points: Decentralization in multi-swarm exploration with hybrid search (MPSO α); Concentration in evaluation and behavior control with diversive curiosity (DC); And their effective combination. According to these, the proposal is expected to alleviate stagnation, and to enhance the search performance.

The rest of the paper is organized as follows. Section 2 briefly reviews the related works on this study. Section 3 introduces basic algorithms of the PSO, EPSO, LRS, and an internal indicator. Section 4 describes the mechanism of the MPSO α /DC and its characteristics. Section 5 analyzes and discusses the experimental results to a suite of five-dimensional (5D) benchmark problems. Finally Sect. 6 gives the conclusions.

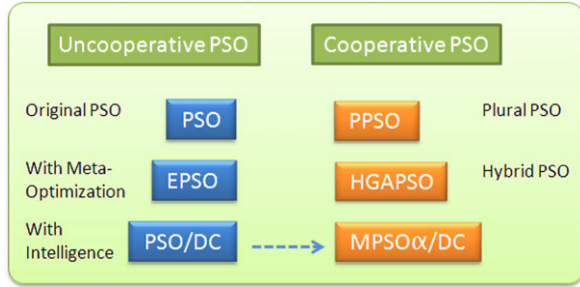
2 Related Works

Since plural particle swarm optimizers are implemented to achieve a certain goal, it is commonly called cooperative PSO. To all sorts of cooperative PSO, El-Abd et al. gave the following definition in [9]:

Multiple swarms (or sub-swarms) searching for a solution (serially or in parallel) and exchanging some information during the search according to some communication strategy.

As mentioned, it is apparent that the core technique of cooperative PSO involves utilizing the dispersiveness in group behaviors to function complement, and relative

Fig. 1 The origin of the MPSO α /DC



information propagation to improve the search accuracy and efficiency. For making more effects, we propose the MPSO α /DC to strengthen the intelligent functions regarding decision making and hybrid search.

Figure 1 illustrates the origin of the MPSO α /DC. As for uncooperative PSO, both of evolutionary particle swarm optimization (EPSO) [28, 29] and particle swarm optimization with diversive curiosity (PSO/DC) [30, 31] were published. The former focuses on obtaining appropriate values of parameters in the PSO by evolutionary computation to rationally manage the trade-off between exploitation and exploration in the PSO heuristic [6]. The latter pays attention to enhance swarm intelligence by introducing diversive curiosity into the PSO.

As a matter of fact, the overall structure of the PSO/DC provides a good framework for methodical expansion. Needless to say, cooperative PSO also belongs to one of optimization methods. The proposed MPSO α /DC is an analogue of the PSO/DC in the curiosity-driven construction. Nevertheless, it has the following outstanding features:

1. Implementing plural particle swarms in parallel to explore for enhancing the adaptability which effectually handles a given optimization problem.
2. Localized random search (LRS) is separately carried out to find the most suitable solution in a small limited space for correcting the solution found by each particle swarm, respectively.
3. Introducing diversive curiosity into the multi-swarm to check the search condition, and to control search behavior for alleviating stagnation.

Obviously, these strategies and the overall composition of the MPSO α /DC are quite different from the parallel particle swarm optimization (PPSO) [3] which only implements plural PSO simultaneously and the hybrid genetic algorithm and particle swarm optimization (HGAPSO) [12] which implements GA and PSO by the mixed operation.

3 Basic Algorithms

This section briefly describes the used PSO, LRS, and an internal indicator.

3.1 The PSO

For the sake of convenience, here, let the search space be N -dimensional, $\Omega \in \mathfrak{R}^N$, the number of particles in a swarm be P , the position of i th particle be $\vec{x}^i = (x_1^i, x_2^i, \dots, x_N^i)^T$, and its velocity be $\vec{v}^i = (v_1^i, v_2^i, \dots, v_N^i)^T$. In the beginning of the PSO search, the particle's position and velocity are generated in random, then they are updated by

$$\begin{cases} \vec{x}_{k+1}^i = \vec{x}_k^i + \vec{v}_{k+1}^i \\ \vec{v}_{k+1}^i = c_0 \vec{v}_k^i + c_1 \vec{r}_1 \otimes (\vec{p}_k^i - \vec{x}_k^i) + c_2 \vec{r}_2 \otimes (\vec{q}_k - \vec{x}_k^i) \end{cases} \quad (1)$$

where c_0 is an inertial coefficient, c_1 is a coefficient for individual confidence, c_2 is a coefficient for swarm confidence. $\vec{r}_1, \vec{r}_2 \in \mathfrak{R}^N$ are two random vectors in which each element is uniformly distributed over $[0, 1]$, and \otimes is an element-by-element multiplication operator. $\vec{p}_k^i (= \arg \max_{j=1, \dots, k} \{g(\vec{x}_j^i)\})$, where $g(\cdot)$ is the criterion value of i th particle at time-step k is the local best position of i th particle up to now, and $\vec{q}_k (= \arg \max_{i=1, 2, \dots} \{g(\vec{p}_k^i)\})$ is the global best position among the whole swarm up to now. In the original PSO, the values of parameter, $c_0 = 1.0$ and $c_1 = c_2 = 2.0$, are set [13].

3.2 The LRS

Random search methods are the simplest ones of stochastic optimization with undirectional search, and are effective in many problems [22]. For obtaining better search performance, we propose to use the LRS to find the most suitable solution from a limited space surrounding the solution found by the PSO. Concretely, the procedure of the LRS is implemented as follows.

- step 1: Let \vec{q}_k^s be a solution found by s th particle swarm at time-step k , and set $\vec{q}_{now}^s = \vec{q}_k^s$. Give the terminating condition, J (the total number of the LRS run), and set $j = 1$.
- step 2: Generate a random data, $\vec{d}_j \in \mathfrak{R}^N \sim N(0, \sigma_N^2)$ (where σ_N is a small positive value given by user, which determines the small limited space). Check whether $\vec{q}_k^s + \vec{d}_j \in \Omega$ is satisfied or not. If $\vec{q}_k^s + \vec{d}_j \notin \Omega$ then adjust \vec{d}_j for moving $\vec{q}_k^s + \vec{d}_j$ to the nearest valid point within Ω . Set $\vec{q}_{new}^s = \vec{q}_k^s + \vec{d}_j$.
- step 3: If $g(\vec{q}_{new}^s) > g(\vec{q}_{now}^s)$ then set $\vec{q}_{now}^s = \vec{q}_{new}^s$.
- step 4: Set $j = j + 1$. If $j \leq J$ then go to step 2.
- step 5: Set $\vec{q}_k^s = \vec{q}_{now}^s$ to correct the solution found by the s th particle swarm at time-step k . Stop the search.

Due to the complementary feature of the hybrid search (i.e. memetic algorithm [17]), the correctional function seems to be close to the HGAPSO in search effect.

3.3 Internal Indicator

Curiosity, as a general concept in psychology, is an emotion related to natural inquisitive behavior for humans and animals, and its importance and effect in motivating search cannot be ignored [7, 19]. Berlyne categorized it into two types: diversive curiosity and specific curiosity [2]. In the matter of the mechanism of the former, Loewenstein insisted that “diversive curiosity occupies a critical position at the crossroad of cognition and motivation” in [15].

Based on the assumption of “cognition” is the act of exploitation, and “motivation” is the intention to exploration, Zhang et al. proposed the following internal indicator for distinguishing the above two behavior patterns [30–32].

$$y_k(L, \varepsilon) = \max \left(\varepsilon - \sum_{l=1}^L \frac{|g(\vec{q}_k^b) - g(\vec{q}_{k-l}^b)|}{L}, 0 \right) \quad (2)$$

where $\vec{q}_k^b (= \arg \max_{s=1, \dots, S} \{g(\vec{q}_k^s)\})$, where S is the number of plural particle swarms) denotes the best solution found by the whole particle swarms at time-step k . L is duration of judgment, and ε is the positive tolerance coefficient (sensitivity). It is evident that the bigger the value of the coefficient ε is, the higher the sensitivity for exploration is, and vice verse.

4 The MPSO α /DC

Figure 2 shows a flowchart of the MPSO α /DC to illustrate the data processing and behavior control in the multi-swarm search. Compared to the PSO/DC [30–32], the most difference (yellow parts) in construction is that plural particle swarms are implemented in parallel, and the LRS is used to correct the solution found by each particle swarm, respectively.

From the whole solutions found by the multi-swarm, the best solution, \vec{q}_k^b , is determined with maximum selection. Then it is put in a solution set for information processing. The mission of the internal indicator is to monitor whether the status of the best solution \vec{q}_k^b continues to change or not. It makes up the concentration in evaluation and behavior control. Concretely, while the value of the output y_k is zero, this means that the multi-swarm is exploring the surroundings of the solution \vec{q}_k^b for “cognition”. If once the value of the output y_k become positive, it indicates that the multi-swarm has lost interest, i.e. feeling boredom, in the solution \vec{q}_k^b for “motivation”.

Due to the reduction of boredom behavior of the multi-swarm search, the search efficiency finding an optimal solution or near-optimal solutions is drastically improved. Here, it is to be noted that the repeat of reinitialization decided by the signal d_k in Fig. 2 is a mere expression style which concretely realizes diversive curiosity to a positive search. Of course, the style is not an isolated one, it also can be implemented by other operation ways in practice.

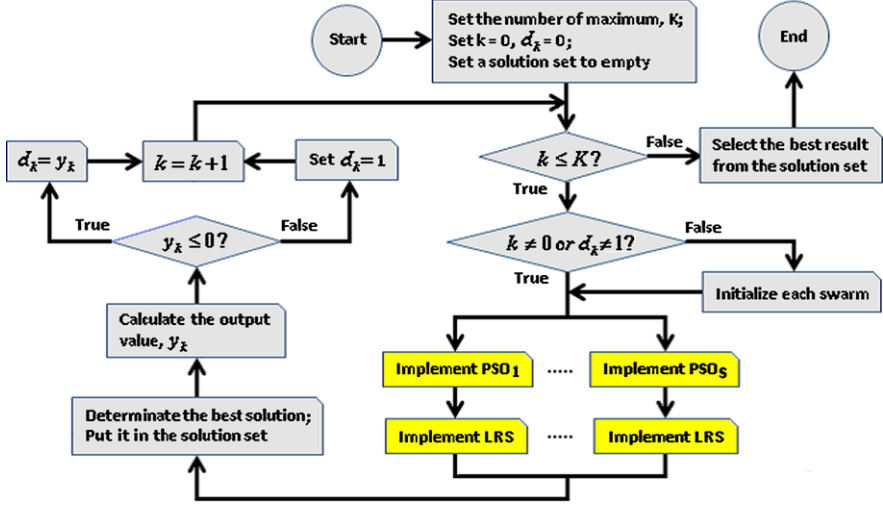


Fig. 2 (Color online) A flowchart of the MPSO α /DC

Table 1 Functions and criteria to the given suite of benchmark problems. The search space for each benchmark problem is limited to $\Omega \in (-5.12, 5.12)^N$

Function	Equation	Criterion
<i>Sphere</i>	$f_{Sp}(\vec{x}) = \sum_{d=1}^N x_d^2$	$g_{Sp}(\vec{x}) = \frac{1}{f_{Sp}(\vec{x})+1}$
<i>Griewank</i>	$f_{Gr}(\vec{x}) = \frac{1}{4000} \sum_{d=1}^N x_d^2 - \prod_{d=1}^N \cos(\frac{x_d}{\sqrt{d}}) + 1$	$g_{Gr}(\vec{x}) = \frac{1}{f_{Gr}(\vec{x})+1}$
<i>Rastrigin</i>	$f_{Ra}(\vec{x}) = \sum_{d=1}^N (x_d^2 - 10 \cos(2\pi x_d) + 10)$	$g_{Ra}(\vec{x}) = \frac{1}{f_{Ra}(\vec{x})+1}$
<i>Rosenbrock</i>	$f_{Ro}(\vec{x}) = \sum_{d=1}^{N-1} [(100(x_{d+1} - x_d^2))^2 + (x_d - 1)^2]$	$g_{Ro}(\vec{x}) = \frac{1}{f_{Ro}(\vec{x})+1}$

In addition, the PSO estimated by EPSO [28, 29] are used in MPSO α /DC for ensuring higher search performance. Specifically, the genetic operations of the EPSO: roulette wheel selection, BLX- α crossover [10], random mutation, non-redundant strategy [27], rank operation, mixing operation, and elitism strategy are adopted [33], and a temporally cumulative fitness function of the best particle is implemented for evaluating the search performance of the PSO.

5 Computer Experiments

To facilitate comparison and analysis of the performance indexes of the proposed method, we use a suite of benchmark problems in Table 1 [24]. And Table 2 gives the major parameters employed for the next experiments.

Table 2 The major parameters used in the EPSO and MPSO α /DC

Parameters	Value	Parameters	Value
The number of individuals, M	10	The number of iterations, K	400
The number of generation, G	20	The maximum velocity, v_{\max}	5.12
Probability of BLX-2.0 crossover, p_c	0.5	The number of particles, P	10
Probability of random mutation, p_m	0.5	The range of LRS, σ_N^2	0.05
The number of particle swarms, S	3	The number of LRS runs, J	10

Table 3 The resulting values of parameters in the PSO for each 5D benchmark problem

Problem	Parameters in the PSO			Fitness
	c_0	c_1	c_2	
<i>Sphere</i>	0.677 ± 0.23	1.129 ± 0.09	0.937 ± 0.65	394.1 ± 0.5
<i>Griewank</i>	0.510 ± 0.26	2.086 ± 0.42	1.025 ± 0.61	396.6 ± 0.6
<i>Rastrigin</i>	1.345 ± 0.54	10.28 ± 3.52	24.92 ± 21.8	395.7 ± 0.5
<i>Rosenbrock</i>	0.902 ± 0.06	1.309 ± 0.56	0.761 ± 0.16	317.1 ± 18.8

5.1 Results of the EPSO

Table 3 shows the resulting values of parameters in the PSO corresponding to each given 5D benchmark problem with 20 trials. We can observe that the average of the parameter values of the estimated PSO are quite different from that of the original PSO for the whole problems. Specially, the average of the parameter values, c_0 , are less than 1 except for the *Rastrigin* problem. This suggests that the active behavior of particles is convergence in exploring solutions. In contrast to this, the average of the parameter values, c_0 , drastically exceeds 1 indicates that the exploration needs to have more randomization without restriction for handling the *Rastrigin* problem.

These estimated PSO in Table 3 as the PSO* are adopted in the MPSO α /DC for ensuring the convergence and search accuracy, and improving the search performance.

5.2 Results of the MPSO* α /DC

For clarifying the characteristics of the MPSO* α /DC, the experiments are carried out by tuning the parameters, L and ε . Figure 3 gives the search performance of the MPSO* α /DC for each problem with 20 trials. From Fig. 3, the following characteristics of the MPSO* α /DC are observed.

- The average of reinitialization frequencies monotonously increases with increment of the tolerance parameter, ε , and decrement of the duration of judgment,

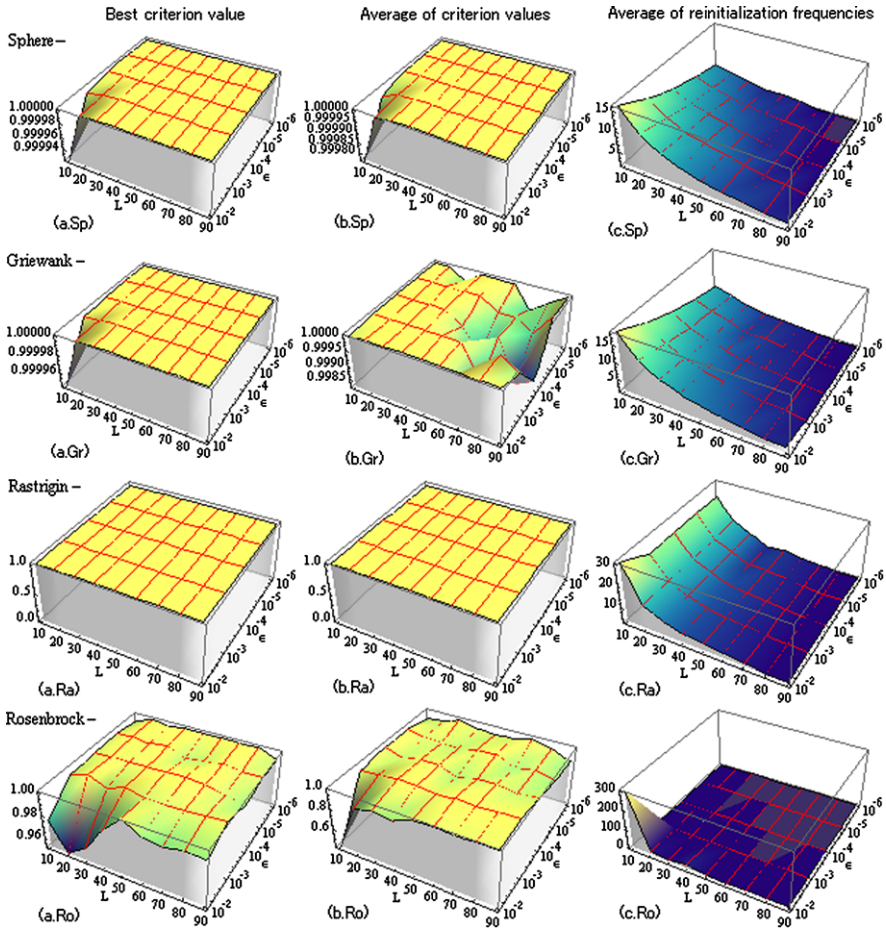


Fig. 3 Distribution of the obtained results with tuning the parameters, L and ε , for each problem. (a) Best criterion value, (b) average of criterion values, (c) average of reinitialization frequencies

L , for each benchmark problem. Whereas, the changes of the best criterion value and the average of criterion values are non-monotonous.

- Both of the best criterion value and the average of criterion values do not change at all with tuning the parameters, L and ε , for the *Rastrigin* problem.
- For obtaining superior search performance of the $MPSO^*\alpha/DC$, the recommended range of the parameters, $L_{Sp}^* \in (30 \sim 90)$ and $\varepsilon_{Sp}^* \in (10^{-6} \sim 10^{-4})$ for the *Sphere* problem; $L_{Gr}^* \in (30 \sim 50)$ and $\varepsilon_{Gr}^* \in (10^{-6} \sim 10^{-4})$ for the *Griewank* problem; $L_{Ra}^* \in (10 \sim 90)$ and $\varepsilon_{Ra}^* \in (10^{-6} \sim 10^{-2})$ for the *Rastrigin* problem; $L_{Ro}^* \in (40 \sim 80)$ and $\varepsilon_{Ro}^* \in (10^{-4} \sim 10^{-3})$ for the *Rosenbrock* problem are available.

As to the *Rastrigin* problem, the resulting best criterion value and the average of criterion values are mostly unchanged with tuning the parameters, L and ε . This result suggests that the optimized PSO* is sufficient to handle the problem.

On the other hand, due to stochastic factor in the PSO search and complexity of the given problems, some irregular change of the experimental results can be discovered in Fig. 3 (b.Gr) and (b.Ro). Moreover, because of the better effect of the hybrid search, the fundamental finding, “the zone of curiosity,” in psychology [7] is not distinguished except for the *Rosenbrock* problem. Hence, the surface of “the average of criterion values” seems to be a plane without the change of the parameters L and ε . This result indicates that the MPSO* α /DC has good adaptability to solve the given problems regardless of that the parameter L is short and ε is bigger.

We also observe that the average of reinitialization frequencies is over 300 in the case of the parameters, $L = 10$ and $\varepsilon = 10^{-2}$, for the *Rosenbrock* problem in Fig. 3 (c.Ro). Since the average of criterion values is the lowest than that in the other cases in Fig. 3 (b.Ro), this result shows that the active exploring of the multi-swarm seems to have entered “the zone of anxiety,” which leads the search performance of the MPSO α /DC to be lower.

5.3 Performance Comparison

To declare the intrinsic characteristics and the effect of the multi-swarm and hybrid search, the following investigation are carried out.

5.3.1 Singular vs. Multiple Swarm Search

For equal treatment in search, the number of particles used in a singular swarm is the same to the total number of particles used in the multi-swarms. Figure 4 shows the resulting difference, $\Delta_{PS} = \bar{g}_P^* - \bar{g}_S^*$ (\bar{g}_P^* : the average of criterion values of the MPSO* α /DC, \bar{g}_S^* : the average of criterion values of the PSO* α /DC). We can see that the search performance of both the MPSO* α /DC and PSO* α /DC seems to be the same for the *Sphere* and *Rastrigin* problems. This is because the simplicity of the *Sphere* problem and the effect of the EPSO to the *Rastrigin* problem. Based on the distribution of positive values of the difference, Δ_{PS} , we also observe that the search performance of the MPSO* α /DC is better than that by the PSO* α /DC for the *Griewank* problem, and for the *Rosenbrock* problem under the condition of $\varepsilon \leq 10^{-5}$.

In other words, under the situation of low-sensitivity, i.e. a feeling of easiness, the search efficiency of plural particle swarms is superior to that by singular particle swarm for each benchmark problem. Accordingly, the effectiveness of implementing plural particle swarms to exploration is demonstrated well.

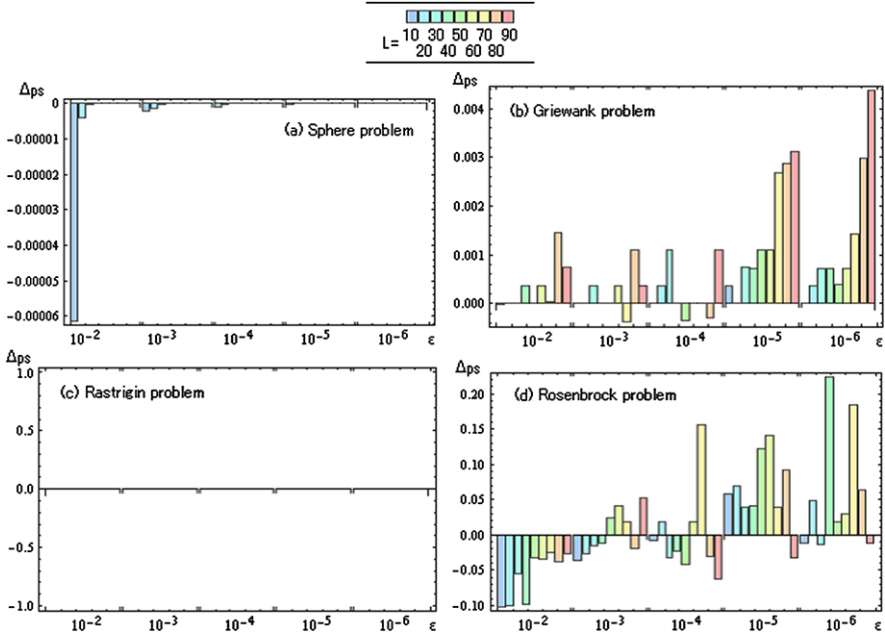


Fig. 4 The performance comparison between the MPSO* α /DC and PSO* α /DC

5.3.2 Effect of the LRS

For investigating the performance difference between the MPSO* α /DC and MPSO*/DC, Fig. 5 shows the obtained experimental results corresponding to same problems. Note that the difference is defined by $\Delta_{PN} = \bar{g}_P^* - \bar{g}_N^*$ (\bar{g}_N^* : the average of criterion values of the MPSO*/DC). Similar to the preceding results, the search performance of both the MPSO* α /DC and MPSO*/DC also seems to be the same for the *Sphere* and *Rastrigin* problems.

On the one hand, the search performance of the MPSO* α /DC is better than that by the MPSO*/DC under the condition of $\epsilon \leq 10^{-5}$ for the *Griewank* problem. On the other hand, the effect of the LRS is not remarkable under the condition of $\epsilon \leq 10^{-5}$ for the *Rosenbrock* problem. This result fits in with “no free lunch” (NFL) theorem [25].

The above results suggest that the effect of the LRS closely depends on the object of search, which related to how to set the parameter values for the running number, J , and the search range, σ_N^2 , and the inherent feature of the given benchmark problems. This is also a hot topic regarding how to rationally manage the trade-off between computational cost and search performance [23]. The details on discussion for the issue are omitted here.

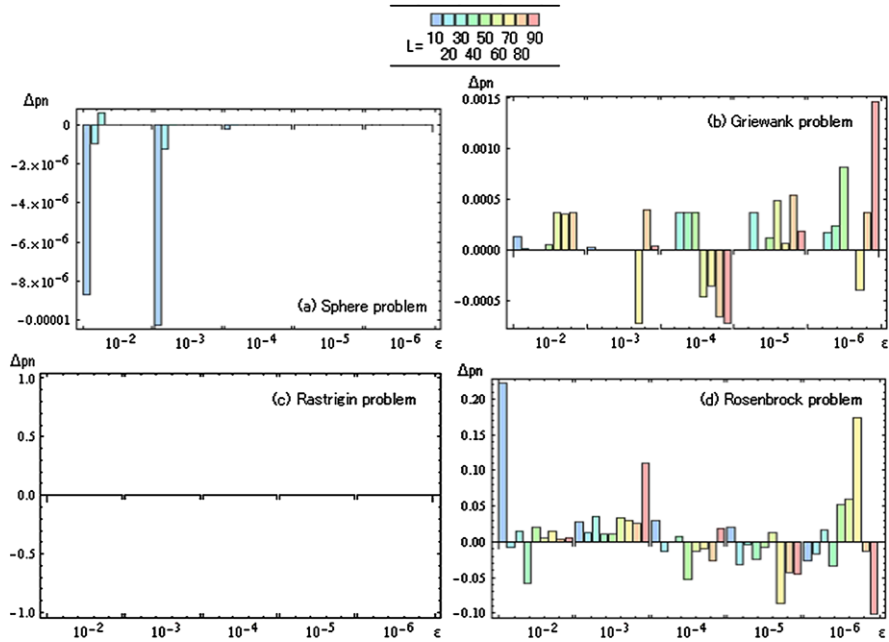


Fig. 5 The performance comparison between the MPSO* α /DC and MPSO*/DC

Table 4 The mean and standard deviation of criterion values in each method for each 5D benchmark problem with 20 trials. The values in bold signify the best result for each problem

Problem	MPSO* α /DC	PSO*/DC	EPSO	OPSO	RGA/E
<i>Sphere</i>	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	0.998 \pm 0.001
<i>Griewank</i>	1.000 \pm 0.000	1.000 \pm 0.000	0.987 \pm 0.010	0.944 \pm 0.043	0.796 \pm 0.117
<i>Rastrigin</i>	1.000 \pm 0.000	1.000 \pm 0.000	1.000 \pm 0.000	0.265 \pm 0.118	0.961 \pm 0.023
<i>Rosenblock</i>	0.989 \pm 0.012	0.625 \pm 0.232	0.469 \pm 0.280	0.392 \pm 0.197	0.372 \pm 0.136

5.3.3 Comparison with Other Methods

For further illuminating the effectiveness of the proposed method, we compare the search performance with the other methods such as the PSO*/DC, EPSO, OPSO (optimized particle swarm optimization) [16], and RGA/E.

Table 4 gives the experimental results of implementing these methods with 20 trials. It is well shown that the search performance of the MPSO* α /DC is better than that by the PSO*/DC, EPSO, OPSO, and RGA/E by comparison with the average of criterion values. The results sufficiently reflect that the merging of both multiple hybrid search and the mechanism of diversive curiosity takes the active role in handling these benchmark problems. In particular, we can confirm that a big increase, i.e. the average of criterion values by implementing the MPSO* α /DC steeply rises

from 0.4694 to 0.9893, in search performance is achieved well for the *Rosenbrock* problem.

6 Conclusions

A newly cooperative PSO – multiple particle swarm optimizers with diversive curiosity, MPSO α /DC, has been proposed in this paper. Owing to the essential strategies of decentralization in search and concentration in evaluation and behavior control, the combination of the adopted hybrid search and the execution of diversive curiosity, theoretically, has good capability, which greatly improves the search efficiency and effectively alleviates stagnation in optimization.

Applications of the MPSO α /DC to a suite of 5D benchmark problems well demonstrated its effectiveness. The experimental results verified that unifying the both characteristics of multi-swarm search and the LRS is successful and effective in convergence and adaptability. Comparison with the search performance of the PSO/DC, EPSO, OPSO, and RGA/E, it is confirmed that the proposed method has an enormous latent capability in handling different benchmark problems and the outstanding powers of multi-swarm search. Accordingly, the basis of the development study of cooperative PSO research in swarm intelligence is expanded and consolidated.

It is left for further study to apply the MPSO α /DC to practical problems in the real-world and dynamic environments.

Acknowledgements This research was partially supported by Grant-in-Aid Scientific Research(C) (22500132) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comput.* **8**(3), 225–239 (2004)
2. Berlyne, D.: *Conflict, Arousal, and Curiosity*. McGraw-Hill Book Co, New York (1960)
3. Chang, J.F., Chu, S.C., Roddick, J.F., Pan, J.S.: A parallel particle swarm optimization algorithm with communication strategies. *J. Inform. Sci. Eng.* **21**, 809–818 (2005)
4. Clerc, M.: *Particle Swarm Optimization*. ISTE Ltd., London (2006)
5. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **6**(1), 58–73 (2000)
6. Cohen, J.D., McClure, S.M., Yu, A.J.: Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philos. Trans. R. Soc. B* **362**, 933–942 (2007)
7. Day, H.: Curiosity and the interested explorer. *Perform. Instruct.* **21**(4), 19–22 (1982)
8. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995*, pp. 39–43 (1995)
9. El-Abd, M., Kamel, M.S.: A taxonomy of cooperative particle swarm optimizers. *Int. J. Comput. Intell. Res.* **4**(2), 137–144 (2008)

10. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms, vol. 2, pp. 187–202. Morgan Kaufman Publishers, San Mateo (1993)
11. Goldberg, D.E.: Genetic Algorithm in Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)
12. Juang, C.-F.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, *IEEE Trans. Syst. Man Cybern. B* **34**(2), 997–1006 (2004)
13. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995, pp. 1942–1948 (1995)
14. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC2002), Honolulu, Hawaii, USA, 12–17 May 2002, pp. 1671–1676 (2002)
15. Loewenstein, G.: The psychology of curiosity: a review and reinterpretation. *Psychol. Bull.* **116**(1), 75–98 (1994)
16. Meissner, M., Schmuker, M., Schneider, G.: Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. *BMC Bioinform.* **7**(125) (2006)
17. Moscato, P.: On evolution, search optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, CA 91125 (1989)
18. Niu, B., Zhu, Y., He, X.: Multi-population cooperation particle swarm optimization. In: LNCS, vol. 3630, pp. 874–883. Springer, Heidelberg (2005)
19. Opdal, P.M.: Curiosity, wonder and education seen as perspective development. *Stud. Philos. Educ.* **20**(4), 331–344 (2001)
20. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization – An overview. *Swarm Intell.* **1**, 33–57 (2007)
21. Shi, Y., Eberhart, R.C.: A modified particle swarm optimiser. In: Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA, 4–9 May 1998, pp. 69–73 (1998)
22. Solis, F.J., Wets, R.J.-B.: Minimization by random search techniques. *Math. Oper. Res.* **6**(1), 19–30 (1981)
23. Spall, J.C.: Stochastic Optimization. In: Gentle, J., et al. (eds.) Handbook of Computational Statistics, pp. 169–197. Springer, Heidelberg (2004)
24. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005. http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/Tech-Report-May-30-05.pdf
25. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
26. Zhang, H.: Multiple particle swarm optimizers with diversive curiosity. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 174–179 (2010)
27. Zhang, H., Ishikawa, M.: A solution to combinatorial optimization with time-varying parameters by a hybrid genetic algorithm. In: Nakagawa, N., et al. (eds.) Brain-Inspired IT I. Int. Congr. Ser., vol. 1269, pp. 149–152. Elsevier, Amsterdam (2004)
28. Zhang, H., Ishikawa, M.: Evolutionary particle swarm optimization (EPSO) – Estimation of optimal PSO parameters by GA. In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2007 (IMECS 2007), Hong Kong, 21–23 March 2007, pp. 13–18 (2007)
29. Zhang, H., Ishikawa, M.: Evolutionary particle swarm optimization – Metaoptimization method with GA for estimating optimal PSO methods. In: Castillo, O., et al. (eds.) Trends in Intelligent Systems and Computer Engineering. LNEE, vol. 6, pp. 75–90. Springer, Heidelberg (2008)

30. Zhang, H., Ishikawa, M.: Improving the performance of particle swarm optimization with diversive curiosity. In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 (IMECS 2008), Hong Kong, 19–21 March 2008, pp. 1–6 (2008)
31. Zhang, H., Ishikawa, M.: Particle swarm optimization with diversive curiosity – An endeavor to enhance swarm intelligence. *IAENG Int. J. Comput. Sci.* **35**(3), 275–284 (2008)
32. Zhang, H., Ishikawa, M.: Characterization of particle swarm optimization with diversive curiosity. *J. Neural Comput. Appl.*, 409–415 (2009)
33. Zhang, H., Ishikawa, M.: The performance verification of an evolutionary canonical particle swarm optimizers. *Neural Netw.* **23**(4), 510–516 (2010)

Predicting the Toxicity of Chemical Compounds Using GPTIPS: A Free Genetic Programming Toolbox for MATLAB

Dominic P. Searson, David E. Leahy,
and Mark J. Willis

Abstract In this contribution GPTIPS, a free, open source MATLAB toolbox for performing symbolic regression by genetic programming (GP) is introduced. GPTIPS is specifically designed to evolve mathematical models of predictor response data that are “multigene” in nature, i.e. linear combinations of low order non-linear transformations of the input variables. The functionality of GPTIPS is demonstrated by using it to generate an accurate, compact QSAR (quantitative structure activity relationship) model of existing toxicity data in order to predict the toxicity of chemical compounds. It is shown that the low-order “multigene” GP methods implemented by GPTIPS can provide a useful alternative, as well as a complementary approach, to currently accepted empirical modelling and data analysis techniques. GPTIPS and documentation is available for download at <http://sites.google.com/site/gptips4matlab/>.

Keywords Genetic programming · Symbolic regression · QSAR · Toxicity

1 Introduction

Genetic programming [6] is a biologically inspired machine learning method that evolves computer programs to perform a task. It does this by randomly generating a population of computer programs (represented by tree structures) and then mutating and crossing over the best performing trees to create a new population. This process is iterated until the population contains programs that (hopefully) solve the task well.

When the task is building an empirical mathematical model of data acquired from a process or system, the GP is often known as symbolic regression. Unlike traditional regression analysis (in which the user must specify the structure of the

D.P. Searson (✉)

Northern Institute for Cancer Research, Newcastle University, Newcastle upon Tyne, UK
e-mail: d.p.searson@ncl.ac.uk

Table 1 Selected chemical engineering applications of GP

Authors	Application area	Model/application
Greeff and Aldrich [2]	Acid pressure leaching	Extent of dissolution model
McKay et al. [9]	Distillation column	Inferential Sensor
Grosman and Lewin [3]	Catalytic reaction	Reaction rate model
Hinchliffe and Willis [4]	Cooking extruder	Dynamic process model
Madar et al. [8]	Polymerisation	Dynamic process model
Wang and Li [17]	Distillation column	Optimise sequence
Seavy et al. [14]	Vapour liquid equilibrium	Hybrid model

model), GP automatically evolves both the structure and the parameters of the mathematical model. Symbolic regression has had both successful academic [1] and industrial applications in a variety of disciplines. For instance, in the discipline of Chemical Engineering, an overview of selected applications of GP is given in the table above.

In all the applications cited in Table 1 there is a common theme; GP is used for symbolic regression and, when using industrial data, the evolved models are shown to have better or comparable accuracy to alternative nonlinear modeling approaches such as neural networks.

The purpose of this chapter is to introduce a free open source MATLAB toolbox called GPTIPS [12, 13] that was written for the specific purpose of performing symbolic regression. GPTIPS employs a unique type of symbolic regression called “multigene” symbolic regression [5] that evolves linear combinations of non-linear transformations of the input variables. When the transformations are forced to be low order (by restricting the GP tree depth) this, in contrast to “standard” symbolic regression, allows the evolution of accurate, relatively compact mathematical models of predictor – response (input – output) data sets, even when there are a large number of input variables. Hence, the authors believe that GPTIPS provides a useful, free and complementary alternative to current data analysis techniques and has a broad spectrum of applicability across many scientific and engineering disciplines.

This chapter is structured as follows. Section 2 provides a brief overview of GP. Next, Sect. 3 discusses the multigene low order GP approach that GPTIPS implements. In Sect. 4, some of the features of GPTIPS are described. In Sects. 5–8, the capabilities of GPTIPS are demonstrated by using it to evolve an accurate, relatively compact mathematical model to predict the toxicity of chemical compounds using a data set from the literature containing over 1000 compounds along with measured toxicity values. Finally, in Sect. 9 we provide some concluding remarks.

2 Genetic Programming

The evolutionary computational (EC) method of GP evolves populations of symbolic tree expressions to perform a user specified task. A comprehensive, free to download introduction to GP and review of the literature is provided by Poli et al. [10] but a brief description of GP is provided here.

In GP, each tree expression can be thought of as being analogous to the DNA of an individual in natural evolution. The evolution of the expressions occurs over a number of generations (iterations) and each new generation of individuals is created from the existing population by direct copying as well as performing operations on the individuals analogous to the alterations to DNA sequences that naturally occur during sexual reproduction and mutation. This is accomplished by evaluating each individual in the current population to determine its ‘fitness’ (i.e. its performance on the user specified objective function or functions) and performing probabilistic selection and recombination of individuals biased towards those that are relatively fit compared to the other individuals in the population.

At the beginning of each run, a population of symbolic expressions is randomly generated. This is accomplished using a simple tree building algorithm that randomly selects nodes, with replacement, from a pool comprising primitive functions (e.g. addition, subtraction, the hyperbolic tangent, natural logarithm, exponential, etc.), the input variables as well as randomly generated constants. These nodes are randomly assembled into tree structured symbolic expressions, subject to user-defined tree size and/or depth constraints. After evolving the population for a number of generations by copying, mutation and recombination operations, the tree expression with the best fitness is usually selected as the best solution to the problem.

Two principal genetic recombination operators are used in GP: sub-tree crossover and sub-tree mutation. Sub-tree crossover is an operation performed on two parent trees that generates two offspring. For each expression, a sub-tree is randomly selected. These sub-trees are then exchanged to create two new expressions to go into the next generation. Sub-tree mutation operates on a single parent expression and generates a single offspring expression. First, a randomly selected sub-tree of the parent is deleted. Then, a new sub-tree is randomly generated using the same tree building algorithm that was used to build the initial population of expressions. The resulting offspring expression is then inserted into the new population. The mutation operation is used relatively infrequently compared to the crossover operation and its purpose is to maintain genetic diversity over the course of the run and to prevent premature convergence to unsatisfactory solutions.

In GP, the choice of the primitive functions is domain dependent and in symbolic regression, where there is little or no prior knowledge of the underlying relationships, mathematical operators such as those described above are typically employed with a high degree of success [9, 13]. In practice, it is often best to perform some initial runs with a few simple primitives (e.g. addition, multiplication and subtraction) and then incrementally add other non-linear primitives – such as the hyperbolic tangent function – to evaluate whether more accurate and compact symbolic expressions may be evolved.

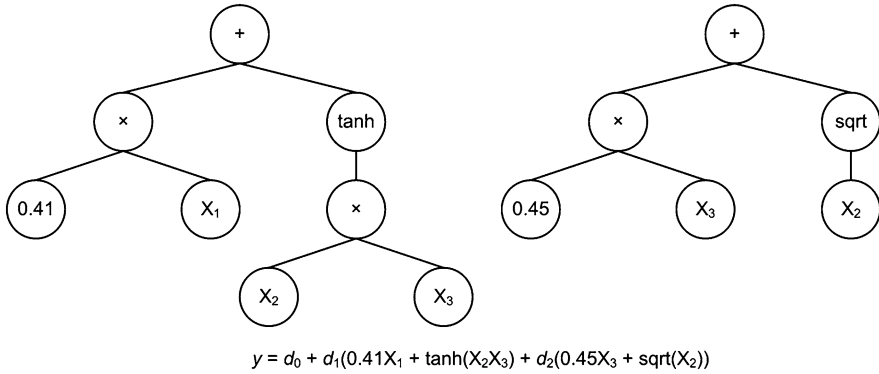


Fig. 1 Example of a multigene symbolic model

3 Multigene Symbolic Regression

Typically, symbolic regression is performed by using GP to evolve a population of trees, each of which encodes a mathematical equation that predicts a $(N \times 1)$ vector of outputs \mathbf{y} using a corresponding $(N \times M)$ matrix of inputs \mathbf{X} where N is the number of observations of the response variable and M is the number of input (predictor) variables. I.e. the i th column of \mathbf{X} comprises the N input values for the i th input variable and may be designated as the input variable x_i .

In contrast, in multigene symbolic regression each symbolic model (and each member of the GP population) is a weighted linear combination of the outputs from a number of GP trees, where each tree may be considered to be a “gene”. For example, the multigene model shown in Fig. 1 predicts an output variable y using input variables X_1 , X_2 and X_3 .

This model structure contains non-linear terms (e.g. the hyperbolic tangent) but is linear in the parameters with respect to the coefficients d_0 , d_1 and d_2 . In practice, the user specifies the maximum number of genes G_{\max} a model is allowed to have and the maximum tree depth D_{\max} any gene may have and therefore can exert control over the maximum complexity of the evolved models. In particular, we have found that enforcing stringent tree depth restrictions (i.e. maximum depths of 4 or 5 nodes) often allows the evolution of relatively compact models that are linear combinations of low order non-linear transformations of the input variables.

For each model, the linear coefficients are estimated from the training data using ordinary least squares techniques. Hence, multigene GP combines the power of classical linear regression with the ability to capture non-linear behaviour without needing to pre-specify the structure of the non-linear model. In Hinchliffe et al. [5] it was shown that multigene symbolic regression can be more accurate and computationally efficient than the standard GP approach for symbolic regression and Searson et al. [13] demonstrated that the multigene approach could be successfully embedded within a non-linear partial least squares algorithm.

In GPTIPS, the initial population is constructed by creating individuals that contain randomly generated GP trees with between 1 and G_{\max} genes. During a GPTIPS

run, genes are acquired and deleted using a tree crossover operator called two point high level crossover. This allows the exchange of genes between individuals and it is used in addition to the “standard” GP recombination operators. If the i th gene in an individual is labelled G_i then a two point high level crossover is performed as in the following example. Here, the first parent individual contains the genes $(G_1G_2G_3)$ and the second contains the genes $(G_4G_5G_6G_7)$ where $G_{\max} = 5$. Two randomly selected crossover points are created for each individual. The genes enclosed by the crossover points are denoted by $\langle \dots \rangle$.

$$(G_1 \langle G_2 \rangle G_3) \quad (G_4 \langle G_5G_6G_7 \rangle)$$

The genes enclosed by the crossover points are then exchanged resulting in the two new individuals below.

$$(G_1G_5G_6G_7G_3) \quad (G_4G_2)$$

Two point high level crossover allows the acquisition of new genes for both individuals but also allows genes to be removed. If an exchange of genes results in an individual containing more genes than G_{\max} then genes are randomly selected and deleted until the individual contains G_{\max} genes.

In GPTIPS, standard GP subtree crossover is referred to as low level crossover. In this case, a gene is selected randomly from each parent individual, standard subtree crossover is performed and the resulting trees replace the parent trees in the otherwise unaltered individual in the next generation. GPTIPS also provides several methods of mutating trees.

The user can set the relative probabilities of each of these recombinative processes. These processes are grouped into categories called events. The user can then specify the probability of crossover events, direct reproduction events and mutation events. These must sum to one. The user can also specify the probabilities of event subtypes, e.g. the probability of a two point high level crossover taking place once a crossover event has been selected, or the probability of a subtree mutation once a mutation event has been selected. However, GPTIPS provides default values for each of these probabilities so the user does not need to explicitly set them.

4 GPTIPS Features

GPTIPS is a predominantly command line driven open source toolbox that requires only a basic working knowledge of MATLAB. A run is configured by a simple configuration M file and there are a number of command line functions to facilitate post-run analyses of the results. Whilst not an exhaustive list, GPTIPS currently contains the following configurable GP features: tournament selection & plain lexicographic tournament selection [7], elitism, three different tree building methods (full, grow and ramped half and half) and six different mutation operators: (1) subtree mutation (2) mutation of constants using an additive Gaussian perturbation (3) substitution of a randomly selected input node with another randomly selected input node (4) set a randomly selected constant to zero (5) substitute a randomly selected constant with

another randomly generated constant (6) set a randomly selected constant to one. In addition, GPTIPS can, without modification in the majority of cases, use nearly any built in MATLAB function as part of the function set for a run. The user can also write bespoke function node M files and fitness functions; hence GPTIPS can be used to solve problems other than non-linear modeling/symbolic regression.

In addition, GPTIPS has a number of features that are specifically aimed at the creation, analysis and simplification of multigene symbolic regression models. These include: (1) use of a ‘holdout’ validation set during training to mitigate the effects of overfitting (2) graphical display of the results of symbolic regression for any multigene model in the final population (3) mathematical simplification of any model (4) conversion to LaTeX format of any model (5) conversion to PNG (portable network graphics) file of the simplified equation of any model (6) conversion of any model to standalone M file for use outside GPTIPS (7) graphical display of the statistical significance of each gene in a model (8) functions to reduce the complexity of any model using “gene knockouts” to explore the trade off of model accuracy against complexity (9) graphical population browser to explore the trade off surface of complexity/accuracy (10) graphical input frequency analysis of individual models or of a user specified fraction of the population to facilitate the identification of input variables that are relevant to the output.

The Symbolic Math toolbox (a commercial toolbox available from the vendors of MATLAB) is required for the majority of the post run simplification and model conversion features and the Statistics Toolbox is required for the display of gene statistical significance. The core functionality of GPTIPS and the ability to evolve multigene models does not, however, require any specific toolboxes.

4.1 Using GPTIPS for Symbolic Regression

An example of a simple configuration file for multiple gene symbolic regression is shown in Fig. 2. It is assumed that data is located in the current directory in the file `mydata.mat` and comprises the training input data variable (`xtrain`), the training output variable (`ytrain`) as well as a testing data set (`xtest` and `ytest`). The data should be arranged by columns, e.g. the n th column of `xtrain` should contain the observations of the n th input variable.

In this example, only a few GPTIPS settings are specified. Any user parameters not explicitly set automatically use the default values. However, the user must at least specify the fitness function, the input and output data and the function nodes to be used in their configuration file. This configuration file first sets population size = 100 and number of generations = 100. The fitness function is then specified (i.e. the name of the M file) as is the fact that this is an error minimisation problem.

Next, the user data file `mydata.mat` is loaded and the variables within this file (here called `xtrain`, `ytrain`, `xtest` and `ytest` but they could be called anything) are assigned. After this is the number of input variables is set as the number of columns in the inputs training data matrix. Next, multigene mode is enabled and the maximum

```
function gp = my_config(gp);  
gp.runcontrol.pop_size = 100;  
gp.runcontrol.num_gen = 100;  
gp.fitness.fitfun = @regressmulti_fitfun;  
gp.fitness.minimisation = true;  
  
load mydata  
gp.userdata.xtrain = xtrain;  
gp.userdata.ytrain = ytrain;  
gp.userdata.xtest = xtest;  
gp.userdata.ytest = ytest;  
  
gp.nodes.inputs.num_inp = size(gp.userdata.xtrain,2);  
  
gp.genes.multigene = true;  
gp.genes.max_genes = 4;  
gp.treedef.max_depth = 5;  
gp.nodes.functions.name = {'times', 'minus', 'plus'};
```

Fig. 2 Example GPTIPS configuration file for multigene symbolic regression

number of genes per individual is set to 4. The maximum tree depth is then set to 5. Finally, the function nodes times, minus and plus are specified.

5 Evolution of a Predictive Model of Aqueous Chemical Toxicity Using GPTIPS

In the remainder of this article we will demonstrate how we have used GPTIPS to evolve a predictive QSAR model of aquatic toxicity for chemical compounds, based on their molecular structure.

QSAR (Quantitative Structure Activity Relationships) is a well established technique for deriving structure property relationships for chemical compounds that can be used to predict the properties of novel chemical structures. Chemical compounds can be represented by a large number of computed numerical values, called “descriptors”, each of which in some way characterises the structure or behaviour of the compound. The idea of QSAR is to build empirical or semi-empirical models that relate the descriptors of a compound to some physical, chemical or biological property. A number of software packages are available to compute descriptor values for compounds with a known structure. Many of these are commercial products (e.g. DRAGON) but there are also free/open source packages e.g. the Chemical Development Kit [15].

A QSAR modelling scenario involves a data set of known chemical compounds and a measured endpoint for each compound. The measured endpoint is the property of interest. Typical properties of interest are those related to pharmaceutical drug development. These include biological activities representing the ability of a drug candidate to perform its desired function (e.g. IC50, the concentration of a compound required to inhibit a particular biological or biochemical function by

half) and the ADME properties (adsorption, distribution, metabolism and excretion) which characterise the behaviour of a of a pharmaceutical drug compound within the organism.

The prediction of chemical toxicity is another chemical property that is of vital importance in both pharmaceutical drug development and managing the environmental risk of chemical compounds. In the latter case there are legal regulatory structures (e.g. the REACH regulations in the European Union – EC 1907/2006) that specify that QSAR models should play a part in managing this risk in order to reduce the costs of experimental toxicity measurement. Hence, the development of effective QSAR modelling methods continues to present a very real and relevant challenge.

There are a number of strategies & protocols for experimentally evaluating chemical toxicity. One commonly accepted method is the measurement of the growth inhibition of ciliated protozoan *T. pyriformis* [18]. There are freely available aquatic toxicity data for more than 1000 compounds, due to the efforts of Schultz and colleagues [11]. Zhu et al. [18] have used this to compile a data set of 1093 unique compounds and have developed a number of predictive QSAR models using various descriptor packages and modelling methodologies. Here, the use of GPTIPS to evolve a predictive model of chemical toxicity using this data set is demonstrated (using the descriptors from the commercial DRAGON package) and the results compared with those published in Zhu et al. [18].

6 Data

The *T. pyriformis* toxicity values (i.e. the response y data) are measured as the logarithm of the 50% growth inhibition concentration $\log(\text{IGC}50^{-1})$. The data available for training QSAR models contains 644 compounds and another 449 compounds are used an external test/validation data set to verify the predictive ability of the models. For each compound 1664 DRAGON descriptor values are used as the predictor data (i.e. the input X data contains 1664 input variables) – compound structures, toxicity and descriptor values are available from the EU CADASTER website at <http://www.cadaster.eu/node/65>. To mitigate against the effects of overfitting, 128 compounds (approximately 20%) in the training data set were randomly selected for use as a holdout validation data set leaving the training data containing 516 compounds. In GPTIPS, holdout validation is performed as follows: at the end of each generation, the “best” individual (as evaluated on the training data) is then evaluated on the holdout validation set. The individual that performs best on the holdout set (over the course of the run) is stored and may be accessed after the run.

7 GPTIPS Run Settings

A GPTIPS run with the following settings was performed: Population size = 500, Number of generations = 500, Tournament size = 12 (with lexicographic selection

pressure), $D_{\max} = 4$, $G_{\max} = 8$, Elitism = 0.01% of population, function node set = {plus, minus, times, tanh, sin}, terminal node set = {1664 DRAGON descriptors $x_1 - x_{1664}$, ephemeral random constants in the range $[-1010]$ }. The default GPTIPS multigene symbolic regression function was used in order to minimise the root mean squared prediction error on the training data.

The following (default) recombination operator event probabilities were used: crossover events = 0.85, mutation events = 0.1, direct reproduction = 0.05. The following sub-event probabilities were used: high level crossover = 0.2, low level crossover = 0.8, subtree mutation = 0.9, replace input terminal with another random terminal = 0.05, Gaussian perturbation of randomly selected constant = 0.05 (with standard deviation of Gaussian = 0.1). These settings are not considered ‘optimal’ in any sense but were based on experience with modelling other data sets of similar size. The run took approximately 15 minutes on a PC with a dual core processor running at 2.2 GHz with 3.5 GB of RAM.

8 Results

The model that performed best on the holdout validation data was chosen. This model has coefficients of determination (i.e. proportion of the variation in the response explained by the model) of R^2 (training) = 0.83, R^2 (holdout) = 0.78 and R^2 (test) = 0.78. In Zhu et al. [18] the results are reported in terms of MAE (mean absolute error) for two test sets referred to in the paper as Validation set 1 (339 compounds) and Validation set 2 (110 compounds) that comprise the whole test set used here. In terms of MAE, the evolved GPTIPS model has MAE(training) = 0.3292, MAE(holdout) = 0.3573 and MAE(test) = 0.3518.

Zhu et al. [18] report the results of a number of individual models, built using various descriptor packages and modelling techniques. Some of these models consider the “applicability domain” (AD) of the compounds (i.e. whether the compounds lie in the region of descriptor space deemed to be suitable for generating a prediction) whereas others do not employ AD considerations. In general, models that consider AD give more accurate predictions but only the results of the non AD models using the DRAGON descriptors are repeated here. The first DRAGON descriptor based model is a support vector machine [16] regression that yields MAE(Validation set 1) = 0.37 and MAE(Validation set 2) = 0.42. This corresponds to an MAE(test) = 0.38. The second DRAGON based model is a k -nearest neighbour (k -NN) approach that achieves MAE(Validation set 1) = 0.29, MAE(Validation set 2) = 0.43 corresponding to MAE(test) = 0.32. Hence it can be seen that the evolved GPTIPS model lies between the SVM and the k -NN approaches, i.e. GPTIPS can achieve predictive performance of the order of the current state of the art empirical modelling methodologies.

GPTIPS was used to mathematically simplify and export the evolved model as a PNG graphics file. This is shown in Fig. 3.

It can be seen that the evolved model is reasonably compact, consists of both linear terms and low order non-linear transformations of the inputs and has selected a small number of descriptors from the 1664 available.

$$\begin{aligned}
y = & -2.092 - 0.7548x_{911} + 0.7548x_{1558} - 0.8997 \tanh(\tanh(x_{1426})) \\
& + 0.09443(x_{911} - x_{1558})x_{654} - 0.1481x_{1552} \\
& + 0.1481x_{391} - 0.2489x_{1429} - 0.2489 \sin(x_{967} - x_{709}) \\
& + 0.7143x_{1245} - 0.5978x_{1662} - 0.5978 \tanh(x_{1429} + x_{525}) \\
& + 0.7802x_{1426} + 0.7802x_{1563}
\end{aligned}$$

Fig. 3 Graphical rendering of evolved symbolic *T. pyriformis* toxicity model

9 Conclusions

In this article we have introduced the multigene symbolic regression capabilities of GPTIPS and demonstrated it with an application in which a predictive symbolic QSAR model of *T. pyriformis* aqueous toxicity was evolved. It was demonstrated that the evolved model is compact and offers similar high performance to recently published QSAR models of the same data. The point of this article is not to assert that multigene symbolic regression (using low order non-linear transforms of the inputs) is better or worse than other methods, but that it is an alternative and complementary approach to existing empirical modelling and data analysis techniques. It is also an approach that is facilitated by the free GPTIPS toolbox for MATLAB, a program that is used widely in academia and industry.

References

1. Alfaro-Cid, E., Esparcia-Alcázar, A.I., Moya, P., Femenia-Ferrer, B., Sharman, K., Merelo, J.J.: Modeling pheromone dispensers using genetic programming. In: Lecture Notes in Computer Science, vol. 5484/2009, pp. 635–644. Springer, Berlin/Heidelberg (2009)
2. Greeff, D.J., Aldrich, C.: Empirical modeling of chemical process systems with evolutionary programming. *Comp. Chem. Eng.* **22**, 995–1005 (1998)
3. Grosman, B., Lewin, D.R.: Automated nonlinear model predictive control using genetic programming. *Comp. Chem. Eng.* **26**, 631–640 (2002)
4. Hinchliffe, M.P., Willis, M.J.: Dynamic systems modelling using genetic programming. *Comp. Chem. Eng.* **27**(12), 1841–1854 (2003)
5. Hinchliffe, M.P., Willis, M.J., Hiden, H., Tham, M.T., McKay, B., Barton, G.W.: Modelling chemical process systems using a multi-gene genetic programming algorithm. In: Genetic Programming: Proceedings of the First Annual Conference (late breaking papers), pp. 56–65. MIT Press, Cambridge (1996)
6. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
7. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), 2002
8. Madar, J., Abonyi, J., Sziefert, F.: Genetic programming for the identification of nonlinear input-output models. *Ind. Eng. Chem. Res.* **44**, 3178–3186 (2005)
9. McKay, B., Willis, M.J., Barton, G.W.: Steady-state modeling of chemical process systems using genetic programming. *Comp. Chem. Eng.* **21**, 981–996 (1997)
10. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk> (2008)
11. Schultz, T.W., Yarbrough, J.W., Woldemeskel, M.: Toxicity to Tetrahymena and abiotic thiol reactivity of aromatic isothiocyanates. *Cell Biol. Toxicol.* **21**, 181–189 (2005)

12. Searson, D.P., Leahy, D.E., Willis, M.J.: GPTIPS: An open source genetic programming toolbox for multigene symbolic regression. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International Multiconference of Engineers and Computer Scientists, IMECS 2010, Hong Kong, 17–19 March 2010
13. Searson, D.P., Willis, M.J., Montague, G.A.: Co-evolution of non-linear PLS model components. *J. Chemom.* **2**, 592–603 (2007)
14. Seavey, K.C., Jones, A.T., Kordon, A.K.: Hybrid genetic programming – First-principles approach to process and product modeling. *Ind. Eng., Chem. Res.* **49**, 2273–2285 (2010)
15. Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., Willighagen, E.: The Chemistry Development Kit (CDK): an open-source Java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci.* **43**, 493–500 (2003)
16. Vapnik, V.N.: *The Nature of Statistical Learning Theory*, second edn. Springer, New York (2000)
17. Wang, X., Li, Y.: Synthesis of multicomponent product separation sequences via stochastic GP method. *Ind. Eng. Chem. Res.* **47**, 8815–8822 (2008)
18. Zhu, H., Tropsha, A., Fourches, D., Varnek, A., Papa, E., Gramatica, P., Oberg, T., Dao, P., Cherkasov, A., Tetko, I.V.: Combinatorial QSAR modeling of chemical toxicants tested against *Tetrahymena pyriformis*. *J. Chem. Inf. Model.* **48**, 766–784 (2008)

Diversity-Driven Self-adaptation in Evolutionary Algorithms

Fanchao Zeng, James Decraene,
Malcolm Yoke Hean Low, Suiping Zhou,
and Wentong Cai

Abstract Pareto-based multi-objective optimization problems (MOPs) are currently best solved using evolutionary algorithms. Nevertheless, the performance of these nature-inspired stochastic search algorithms still depends on the suitability of their parameter settings with respect to specific optimization problems. The tuning of the parameters is a crucial task which concerns resolving the contrary goals of convergence and diversity. To address this issue, we propose a diversity-driven self-adaptive mechanism (SAM) for the simulated binary crossover. This novel technique exploits and optimizes the balance between exploration and exploitation during the evolutionary process. This “explore first and exploit later” approach is addressed through the automated and dynamic adjustment of the distribution index of the simulated binary crossover (SBX) operator. We conducted a series of experiments where SAM is applied to the Non-dominated Sorting Genetic Algorithm to solve the Sphere, Rastrigin, and ZDT optimization problems. Our experimental results have shown that our proposed self adaptation mechanism can produce promising results for both single and multi-objective problem sets.

Keywords Self-adaptive · Parameter tuning · Simulated binary crossover · Evolutionary algorithm

1 Introduction

Evolutionary algorithms can efficiently solve multi-objective optimization problems (MOPs) by obtaining diverse and near-optimal solution sets. Multiple evolutionary techniques have been proposed for MOPs. Among them, Non-dominated Sorting

F. Zeng (✉)

Parallel and Distributed Computing Centre, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore
e-mail: fczeng@ntu.edu.sg

Genetic Algorithms II (NSGA-II) [6] and Strength Pareto Evolutionary Algorithm II (SPEAII) [13] are commonly regarded as the state-of-the-art multi-objective evolutionary algorithms (MOEAs).

In MOEAs, crossover and mutation operators are typically utilized to produce offspring solutions from selected parent individuals. Both operators involve parameters which dictate:

1. The frequency (crossover and mutation rate) of the evolutionary operations.
2. The spread (crossover and mutation distribution index) of offspring solutions.

Both the frequency and spread properties govern the conflicting convergence and diversity dynamics of the evolutionary process. Consequently, the performance of MOEAs depends on the suitability of the above parameters setting with respect to specific optimization problems. The tuning of these parameters is thus a critical time-consuming optimization process. As a result, this limits the applicability of MOEAs to provide decision support for real life problems. To address this issue, we propose a novel self-adaptive mechanism (SAM) which aims at improving the MOEA's performance (when applied to different optimization problems) through automatically adjusting/balancing the exploration and exploitation of candidate solutions during the evolutionary search. SAM can dynamically adjust the distribution index of SBX operator in NSGA-II. Identifying a suitable distribution index (η_c) enables NSGA-II to optimize the balance between exploration and exploitation during the different stages of the evolutionary search.

The essential idea of SAM is that if the diversity running performance is poor, strong evolutionary operation should be applied to break the clusters of candidate solutions and vice versa. Also, the crowding distance is an estimate of the surrounding density of a given solution point and it could be regarded as a criterion to determine the value of this solution. Hence, if the crowding distance is relatively high, soft evolutionary operation is required to preserve the solution points.

The remainder of the paper is structured as follows: A description of related work is first presented. This is followed with an introduction to the SBX operator and diversity running performance metric. Then, a detailed description of the self-adaptive mechanism is provided. A series of experiments involving single and multi-objective optimization problems are conducted and discussed. Our conclusion and future work are then finally outlined.

2 Related Work

Past studies [7, 11] have proven the efficiency of the “explore first and exploit later” concept which relies on the intensive exploration of candidate solutions during the early stage and local fine-tuning during the later/final stage of the search.

To exploit this concept in MOEAs, several self-adaptation approaches have been proposed [1, 10, 11]. Utilizing the feedback from the search, several adaptive parameter control mechanisms were used to obtain a smooth navigation over the search space. For instance, in [1] a self-adaptive Pareto Differential Evolution

(PDE) algorithm was proposed which self-adapts the crossover and mutation rate. A deterministic-scheduled decreasing mutation rate was defined in [11] which also implemented an adaptive variation operator that facilitated the exchange of search information in MOPs. These self-adaptation approaches demonstrated significant improvements over static counterparts; note that these methods focused on the effects of changing the crossover/mutation rates (i.e., frequency) instead of the distribution index parameter (i.e., spread). The detailed taxonomy of self-adaptation of the crossover/mutation rates can be found in [9]. Here we propose a complementary investigation examining the effects of the spread property.

To our knowledge, the only significant reported study addressing spread was carried out in [5], in which a self-adaptive SBX (SA-SBX) was introduced to dynamically adjust (at each generation) the distribution index of SBX in NSGA-II. SA-SBX was found to produce better results on both single and multiple objective optimization problems compared to the SBX with fixed value of the distribution index. Nevertheless, a drawback of SA-SBX is that it requires another critical user-predefined parameter α . According to the experiments reported in [5], SA-SBX would outperform the traditional non self-adaptive counterpart only when α is “well tuned” manually. Although the Deb et al.’s approach demonstrated better performances, their method introduced an additional difficulty in the already complex parameter tuning process. Consequently such approaches do not resolve the robustness and applicability issues of MOEAs.

In contrast with Deb et al.’s approach, we propose a self-adaptive method which does not introduce another critical parameter to be predefined by the user. This self-adaptive mechanism is presented in the next section.

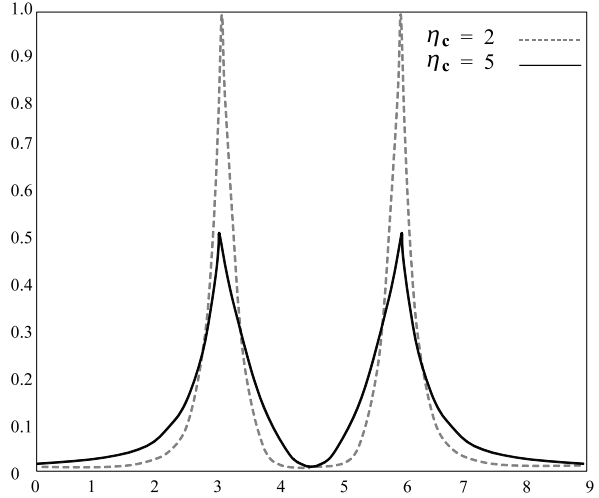
3 Self-adaptive Mechanism

The working principles of SBX are described to emphasize the importance of distribution index η_c in generating the offspring solutions. Then, the implementation details of the diversity running performance metric are presented and the concept of crowding distance is introduced. Finally, we present the self-adaptive mechanism (SAM) which can dynamically adjust the distribution index in SBX using the feedback information from both the diversity running performance metric and the crowding distance.

3.1 Simulated Binary Crossover (SBX)

The SBX crossover operator [2] creates two offspring solutions (represented as real values) from two selected parent solutions. The procedure of deriving offspring solutions $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from the parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$ is as follow.

Fig. 1 The probability density function for creating offspring solutions with the SBX operator (adapted from [2])



A random number $u \in [0, 1]$ is generated. Given a pre-specified probability distribution function (Eq. 1), the value of β_i (mathematical definition of β_i , see Eq. 8) is determined so that the area under the probability curve from zero to β_i is equal to u . The distribution index η_c is a non-negative real number. Figure 1 illustrates the probability density function for creating offspring solutions using the SBX operator from two example parents $x_i^{(1,t)} = 3$ and $x_i^{(2,t)} = 6$ with distribution index of $\eta_c = 2.0$ and $\eta_c = 5.0$. Larger values of η_c are more likely to produce “near parent” solutions whereas smaller values of η_c lead to a more diverse search. After obtaining β_i from Eq. 2, the offspring solutions are calculated using Eqs. 3 and 4.

$$f(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \beta_i \leq 0.5 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise} \end{cases} \quad (1)$$

$$\beta_i = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & u \leq 0.5 \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta_c+1}}, & u > 0.5 \end{cases} \quad (2)$$

$$x_i^{(1,t+1)} = 0.5[(1 + \beta_i)x_i^{(1,t)} + (1 - \beta_i)x_i^{(2,t)}] \quad (3)$$

$$x_i^{(2,t+1)} = 0.5[(1 - \beta_i)x_i^{(1,t)} - (1 + \beta_i)x_i^{(2,t)}] \quad (4)$$

3.2 Diversity Running Performance Metric

A modified diversity running performance metric is implemented to dynamically assess the diversity performance of the generated solution sets. This diversity running performance metric is based on the running performance metrics proposed in [3]. Two principal modifications are introduced:

Table 1 Mapping table to assign a value to $m()$ (adapted from [3])

$h(i-1)$	$h(i)$	$h(i+1)$	$m(h(i-1), h(i), h(i+1))$
0	0	0	0.00
0	0	1	0.50
1	0	0	0.50
0	1	1	0.67
1	1	0	0.67
0	1	0	0.75
1	0	1	0.75
1	1	1	1.00

1. The number of grids (approximating the diversity of the population, see Fig. 2) is derived by dividing the population size by the number of objectives (instead of requiring the user to manually define it).
2. Deb et al.'s approach is limited by the requirement of a priori knowledge of the target solutions distribution. Using this information, the number of grids can be determined/fitted. Nevertheless in real life optimization problems, this information is usually unavailable. Here the running metric does not refer to any pre-specified target set of solution points. Instead the running metric is employed to converge towards an ideal target set of solutions where each grid would possess a representative solution point.

Given the minimal and maximal boundary values, the hyperplane is thus divided into a number of grids (population size divided by the number of objectives). The diversity performance metric is based on whether each grid contains a solution point or not. The best diversity performance is achieved if all grids contain at least a solution point. The steps to calculate the diversity are as follows.

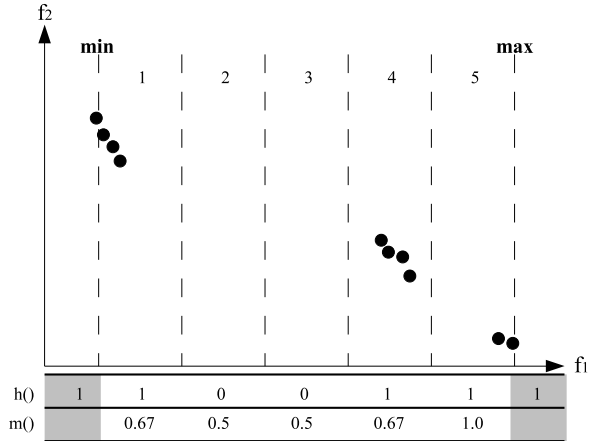
- Step 1: Calculate diversity array. The number of integer variables in the diversity array is equal to the number of grids in the hyperplane. Each variable in the diversity array corresponds to one particular grid i . The value $h(i)$ of the i th elements is derived using Eq. 5.

$$h(i) = \begin{cases} 1, & \text{if grid } i \text{ contains a representative point} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

- Step 2: Assign a value, $m()$ to each grid i depending on its neighboring grids $h()$ values in the diversity array. The value of the i th grid is calculated as shown in Table 1.

For example let us consider the grid patterns $p_1 = 010$ (i.e., $h(i-1) = 0$, $h(i) = 1$ and $h(i+1) = 0$ and $p_2 = 101$). According to Table 1, we obtain $m(p_1) = m(p_2) = 0.75$ which represent a good periodic spread pattern. Whereas if we consider $p_3 = 110$, we obtain $m(p_3) = 0.67$ meaning that p_3 covers a smaller spread.

Fig. 2 Example of computing the diversity metric



- Step 3: For each objective, calculate the diversity measure d_m by averaging the $m()$ values.

$$d_m = \frac{\sum_i^{\text{number of grids}} m(h(i-1), h(i), h(i+1)))}{\text{Number of Grids}} \quad (6)$$

To illustrate the procedure to calculate the diversity measure, an example is presented in Fig. 2.

In this example, a two-objective (f_1 and f_2) minimization problem is examined. The solution points are marked as points. The $f_2 = 0$ plane is used as the reference plane. Suppose the population size is 10, we divide the range of f_1 values into $10/2 = 5$ grids. Then, for each grid, the value of $h()$ is calculated based on whether the grid contains a representative solution point or not. Then, the value of $m()$ and the diversity measure are calculated based on a sliding window containing three consecutive grids. The $h()$ values of the imaginary boundary grids are always 1 as shown in the shaded grids.

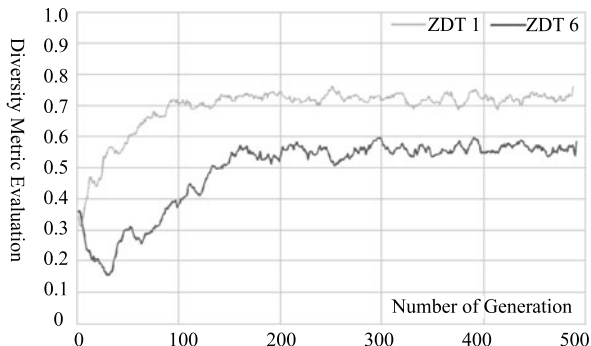
$$d_m(f_1) = \frac{0.67 + 0.50 + 0.50 + 0.67 + 1}{5} = 0.668$$

- Step 4: Calculate overall diversity performance metric by averaging the diversity measures of all objective spaces.

$$\text{Diversity Metric} = \frac{\sum_i^{\text{Number of Objectives}} d_m(i)}{\text{Number of Objectives}} \quad (7)$$

Figure 3 illustrates the running diversity metric obtained from two example experimental runs. For ZDT1, after the 100th generation, the diversity metric oscillates around a value of 0.85. In ZDT6 case, this diversity metric reaches steady state after 160 generations. Similar observations have been reported in [8]. In our implementation, this diversity running performance metric is used to return feedback about the search space. Once the diversity metric stabilizes (i.e., when the exploration phase terminates) the exploitation phase may initiate.

Fig. 3 Diversity metric dynamics for ZDT1 and ZDT6 using NSGA-II from two example experimental runs using the following parameters: population size = 100, crossover distribution index $\eta_c = 20.0$, mutation distribution index $\eta_m = 50.0$, crossover probability $p_c = 1.0$, mutation probability $p_m = 1/30$ and $1/10$ for the benchmark problems ZDT1 and ZDT6 respectively, and maximum number of generations $g = 500$



3.3 The Crowding Distance

The crowding distance indicator was proposed in [6]. It serves as an estimation of the size of the largest cuboid enclosing the solution point.

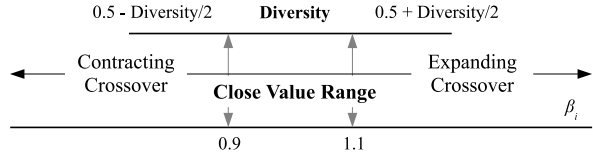
It could be regarded as a criterion to determine the value of the solution point. In this scheme, “boundary solutions” or highest and lowest objectives are given the maximum value in order to retain them. The crowding distance can be calculated by measuring the distance between the two immediate neighbors of a given point along each of the objective dimensions. Lastly, the “final crowding distance” is computed by adding the crowding distances obtained for each objective.

3.4 Diversity-Driven Self-adaptive SBX

In most applications of NSGA-II, the crossover and mutation distribution index η_c and η_m are fixed. Specifically, a fixed value of $\eta_c = 2.0$ is typically chosen for single-objective optimization problems [4] whereas $\eta_c = 20.0$ is commonly used for ZDT benchmark problem sets. Although using a fixed value of η_c can also lead to the implementation of self-adaptive techniques, past studies using the SBX operator with fixed distribution index could not solve multi-modal problems such as the Rastrigin’s function [2].

We suggest a self-adaptive mechanism to dynamically update η_c . Here we assume that for MOPs, the optimal diversity performance could only be achieved when the solution set is close to the optimal solution set. Hence, if optimal diversity performance is achieved, the distribution index η_c should be large enough to make the offspring solutions very similar to their parents. On the other hand, if the diversity performance is poor, strong crossover operation should be applied to break the clusters of solution points. In the beginning stage of the search process,

Fig. 4 Mapping between β_i and u value in SAM



relatively low diversity metric results in strong crossover operation to explore the search space and in the later stage, soft crossover operation is applied to exploit local near-optimal solutions. Thus, this diversity-driven SAM can effectively exploit the concept of “explore first and exploit later”. Also, a large crowding distance means that the surrounding density of the solution point is low, consequently soft crossover operation should be applied to preserve it.

The above SAM algorithm is now detailed:

- Step 1: Calculate the diversity running performance metric.
- Step 2: Derive the reference crossover distribution index η_c based on the diversity performance.

The spread β_i of the offspring solution points with respect to the parent points is obtained in Eq. 8. Based on β_i , crossover can be classified into three classes, namely contracting crossover ($\beta_i < 1$), stationary crossover ($\beta_i = 1$), and expanding crossover ($\beta_i > 1$). The expanding crossover can “expand” the parent points to form more diverse offspring points. Contracting crossover has the opposite effect of contracting the parent points. We define the value range of β_i from 0.9 to 1.1 as the close value range (CVR) where the generated offspring solutions are considered very similar to parent solutions. This range was determined based on parametric studies carried out in [12].

$$\beta_i = \left| \frac{x_i^{2,t+1} - x_i^{1,t+1}}{x_i^{2,t} - x_i^{1,t}} \right| \quad (8)$$

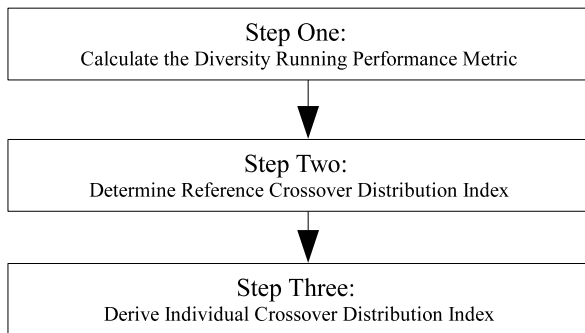
Here we determine the reference distribution index η_c such that the probability of β_i falling into the CVR (i.e., $\beta_i \in [0.9, 1.1]$) equals to the diversity performance metric as illustrated in Fig. 4. For example, if the diversity running performance metric is 0.70, then we should make sure that 70% of the time $\beta_i \in [0.9, 1.1]$. By mapping the random number u to β_i (using Eq. 2), we have $u \in [0.15, 0.85]$. Then η_c can be calculated using Eq. 9:

$$\eta_c = \begin{cases} \frac{\log 2u}{\log \beta_i} - 1, & \text{if } \leq 0.5 \\ -(1 + \frac{\log 2(1-u)}{\log \beta_i}), & \text{otherwise} \end{cases} \quad (9)$$

The distribution indexes $\eta_c = 10.42$ and $\eta_c = 11.63$ are averaged and we obtain a reference crossover distribution index $\eta_c = 11.0$ to produce offspring solutions.

Randomly initialized population causes poor diversity performance at the beginning and consequently lowers the probability of β_i falling into CVR. In the later stage, the diversity performance stabilizes at a relatively higher value and the exploitation phase starts as the probability of β_i in-between CVR is higher.

Fig. 5 Flowchart of the self-adaptive mechanism for SBX



- Step 3: According to the crowding distances cd of the selected parents, individual crossover distribution indexes are assigned to improve the efficiency and accuracy of the crossover operator.

For each generated offspring solution, individual crossover indexes are computed using the expression below.

$$\eta_{\tilde{c}} = \eta_c \times \frac{cd_1 + cd_2}{2 \times \overline{cd}} \quad (10)$$

where cd_1 and cd_2 are the crowding distances of the two selected parents and \overline{cd} is the average crowding distance of the entire population. As devised in the crowding distance scheme, the boundary solutions have maximum values. Consequently these values are not included in the calculation of the average crowding distance. Instead, offspring solutions having boundary solutions as parent points are assigned with the highest distribution index to retain them. Following the previous example, $\eta_c = 11.0$ and the crowding distances of the two parents of offspring solution are 0.65 and 0.95 respectively with an average crowding distance of 0.50. Given Eq. 10, we have: $\eta_{\tilde{c}} = 11.0 \times \frac{1.60}{1.00} = 17.6$.

The above step are summarized in Fig. 5.

4 Empirical Results

We now apply the self-adaptive mechanism in NSGAI1 to two single-objective functions (Sphere and Rastrigin's Function) and 5 multi-objective benchmark problems (ZDT1, 2, 3, 4 and 6). For single-objective optimization problems, a fixed value of $\eta_c = 2.0$ is selected as suggested in [2]. The following parameter setting is used for multi-objective benchmarks: $\eta_c = 20.0$, $\eta_m = 50.0$, $p_c = 1.0$, $p_m = 1/(\text{number of variables})$. Each set of experiments (where 20,000 and 100,000 fitness evaluations are conducted for single and multi-objective benchmark problems respectively) is repeated ten times.

4.1 Single-Objective Functions

$$f(x) = \sum_{i=1}^n x_i^2, \quad x_i \in [-5.12, 5.12] \quad (11)$$

Firstly, NSGAI with fixed value of $\eta_c = 2.0$ SBX operator is employed on the 30-variable sphere function (Eq. 11) with following parameter settings: population size = 100, $p_c = 1$, $p_m = 1/30$, and $\eta_m = 50$. Next, SAM is embedded in NSGAI to dynamically update η_c . SAM can obtain better global minimal value of 0.00598655 as compared to 0.0100824 in NSGAI with fixed η_c .

$$f(x) = \sum_{i=1}^n (x_i^2 + 10(1 - \cos(2\pi x_i))), \quad x_i \in [-5.12, 5.12] \quad (12)$$

Then, we investigate the Rastrigin's function (Eq. 12) which contains many local optima and one global minimum ($x_i = 0, i = 1, 2, 3, \dots, n$). This function is difficult to solve for global optimality using the real-coded genetic algorithm, especially when the initial population does not bracket the global optimum [4]. The experimental settings are the same as that of previous one for sphere function. SAM can obtain better global minimal value of 17.8358 as compared to 40.4383 in NSGAI with fixed η_c .

4.2 Multi-objective Functions

Two benchmark metrics, Inverted Generational Distance (IGD) and SPREAD are employed to measure the performance. IGD uses the true Pareto front as a reference and measure the distance of each of the solution points with respect to the front as Eq. 13:

$$IGD = \frac{\sqrt{\sum_i^N d_i^2}}{N} \quad (13)$$

where d_i is the Euclidean distance between the solution points and the closest member of the true Pareto front. N is the number of solution points in the true Pareto front. When $IGD = 0$, it indicates that the obtained solution set is in the true Pareto front. The SPREAD indicates the extent of spread among the obtained solutions and is computed as follows.

$$Spread = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (14)$$

where d_f and d_l are the Euclidean distances between the boundary solutions (of the obtained solution set). d_i is the Euclidean distance between consecutive solution points. Tables 2 and 3 summarize the experimental results.

Table 2 Results for the inverted generational distance metric between SAM NSGA-II and NSGA-II

	Inverted generational distance (IGD) metric			
	NSGA-II with SAM		NSGA-II	
	Mean	Standard deviation	Mean	Standard deviation
ZDT1	1.74E-04	5.10E-06	1.91E-04	1.08E-05
ZDT2	1.79E-04	5.64E-06	1.88E-04	8.36E-06
ZDT3	2.46E-04	7.74E-06	2.59E-04	1.16E-05
ZDT4	1.67E-04	8.02E-06	1.84E-04	9.86E-06
ZDT6	1.51E-04	1.06E-05	1.59E-04	1.24E-05

Table 3 Results for the Spread diversity metric between SAM NSGA-II and NSGA-II

	Spread diversity metric			
	NSGA-II with SAM		NSGA-II	
	Mean	Standard deviation	Mean	Standard deviation
ZDT1	2.92E-01	3.25E-02	3.83E-01	3.14E-02
ZDT2	3.15E-01	2.01E-02	3.52E-01	7.25E-02
ZDT3	7.31E-01	1.20E-02	7.49E-01	1.49E-02
ZDT4	3.27E-01	2.71E-02	3.96E-01	2.94E-02
ZDT6	4.73E-01	2.98E-02	4.80E-01	4.49E-02

As observed in Tables 2 and 3, SAM achieved lower means for both IGD and Spread diversity metrics in all ZDT problem sets compared to NSGA-II with fixed distribution index. Note that no prior parameter-tuning was conducted for the runs using SAM.

5 Conclusion

Utilizing the feedback from diversity running performance metric and the crowding distance, a self-adaptive mechanism was suggested to dynamically adjust the distribution index of the SBX operator. SAM is able to exploit and control the balance between exploration and exploitation during the different evolutionary search stages. We demonstrated that SAM can effectively alleviate the tedious process of parameter tuning which is a time-consuming trial-and-error optimization process. On several benchmark problem sets, SAM was found to outperform NSGA-II with fixed distribution index. Further investigations are needed to evaluate SAM when applied to real-time problems where the Pareto front may be dynamic. Finally, SAM will also be implemented and evaluated in other MOEAs such as the Bee Colony Optimization and Artificial Immune System techniques.

References

1. Abbass, H.A.: The self-adaptive Pareto differential evolution algorithm. In: Congress on Evolutionary Computation (CEC'2002), vol. 1, pp. 831–836 (2002)
2. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Syst.* **9**(2), 115–148 (1995)
3. Deb, K., Jain, S.: Running performance metrics for evolutionary multi-objective optimization. KanGAL Report, 2002004 (2002)
4. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. *Evol. Comput.* **10**(4), 371–395 (2002)
5. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, p. 1194. ACM, New York (2007)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
7. Janikow, C.Z., Michalewicz, Z.: An experimental comparison of binary and floating point representations in genetic algorithms. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 31–36. Morgan Kaufmann Publishers, San Mateo (1991)
8. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms. TIK Report, 108 (2001)
9. Meyer-Nieberg, S., Beyer, H.G.: Self-adaptation in evolutionary algorithms. In: Parameter Setting in Evolutionary Algorithms, pp. 47–75 (2007)
10. Tan, K.C., Chiam, S.C., Mamun, A.A., Goh, C.K.: Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization. *Eur. J. Oper. Res.* **197**(2), 701–713 (2009)
11. Tan, K.C., Goh, C.K., Yang, Y.J., Lee, T.H.: Evolving better population distribution and exploration in evolutionary multi-objective optimization. *Eur. J. Oper. Res.* **171**(2), 463–495 (2006)
12. Zeng, F., Low, M.Y.H., Decraene, J., Zhou, S., Cai, W.: Self-adaptive mechanism for multi-objective evolutionary algorithms. In: Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 7–12 (2010)
13. Zitzler, E., Laumanns, M., Thiele, L., et al.: SPEA2: improving the strength Pareto evolutionary algorithm. In: EUROGEN, pp. 95–100 (2001)

A New Rearrangement Plan for Freight Cars in a Train

Q-Learning for Minimizing the Movement Counts of Freight Cars

Yoichi Hirashima

Abstract In this paper, a new Q-Learning method for transfer scheduling of freight cars in a train is proposed. In the proposed method, the number of freight-movements in order to line freights in the desired order is reflected by corresponding evaluation value for each pair of freight-layout and removal-destination at a freight yard. Evaluation values are obtained by the Q-Learning method. The best transfer scheduling can be derived by selecting the removal-action of freight that has the best evaluation value at each freight-layout.

Keywords Scheduling · Container transfer problem · Q-Learning · Freight train · Marshaling

1 Introduction

In recent years, logistics with freight train has important role in ecological aspects, because railway logistics is known to have smaller environmental load as compared to goods transportation with trucks [7]. A freight train consists of several railway cars, and each car has one or several containers. Commonly, goods are packed into containers and each container in a freight train has its own destination. Since freight trains can transport goods only between railway stations, modal shifts are required for delivering them to area that has no railway. In intermodal transportations from the road to the rail, containers carried into the station are loaded on freight cars in the arriving order. The initial layout of freight cars is thus random. For efficient shift, the desirable layout should be determined considering destination of container. Then, freight cars must be rearranged before jointing to the freight train.

Y. Hirashima (✉)

Osaka Institute of Technology, 1-79-1, Kitayama, Hirakata, Osaka, 573-0196, Japan

e-mail: hirash-y@is.oit.ac.jp

In general, the rearrangement process is conducted in a freight yard that consists of a main-track and several sub-tracks. Freight cars are initially placed on sub-tracks, then, rearranged and lined into the main track. Although similar problems are treated by mathematical programming and genetic algorithm [1–3, 5, 6, 8], the total number of movements of freight cars is not directly evaluated for realistic problems. Recently, a reinforcement learning method to improve marshaling plan based on the number of movements of freight cars has been proposed [4]. The method can obtain the optimal solution for simple cases by autonomous learning.

In this paper, a new scheduling method is proposed in order to rearrange and line freight cars by the desirable order onto the main track. In the proposed method, the focus is centered on to reduce the number of car-movements that achieves desirable order on the main track. The optimal layout of freight cars in the main track is derived based on the destination of freight cars. This yields several desirable layouts of freight cars in the main track, and the optimal layout that can achieve the smallest number of car-movements is obtained by autonomous learning. Simultaneously, the optimal sequence of car-movements that can achieve the desired layout is obtained by autonomous learning. Also, the feature is considered in the learning algorithm. The learning algorithm is derived based on the Q-Learning [9], which is known as one of the well established realization algorithm of the reinforcement learning.

In the learning algorithm, the state of sub-tracks is defined by using a layout of freight cars, the car to be moved, and the destination of the removed car. An evaluation value called Q-value is assigned to each state, and the evaluation value is calculated by several update rules based on the Q-Learning algorithm. Update rules are independent to each other and the Q-value in one update rule is referred from another update rule, so that Q-values are discounted according to the number of car-movements inside the sub-tracks. Consequently, Q-values at each state represent the total number of car-movements inside sub-tracks required to achieve the best layout from the state. Update rules of the proposed method are derived based on the method in [4] and modified by excluding the evaluation for car-movements between sub-tracks and the main track in order to improve learning performances. Moreover, in the proposed method, only referred Q-values are stored by using table look-up technique, and the table is dynamically constructed by binary tree in order to obtain the best solution with feasible memory space.

In order to show effectiveness of the proposed method, computer simulations are conducted for two cases including a simple problem and a relatively complex problem.

2 Problem Description

The yard consist of 1 main track and m sub-tracks. Define k as the number of freight cars placed on the sub-tracks, and they are carried to the main track by the desirable order based on their destination. In the yard, a locomotive moves freight cars from sub-track to sub-track or from sub-track to main track. The movement of freight

Fig. 1 Freight yard

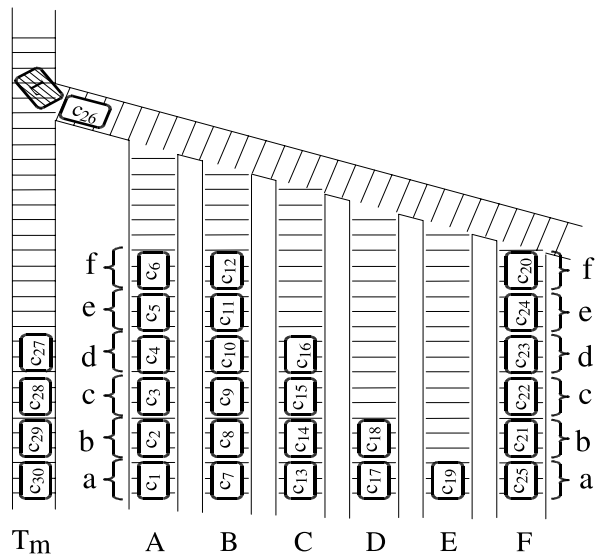
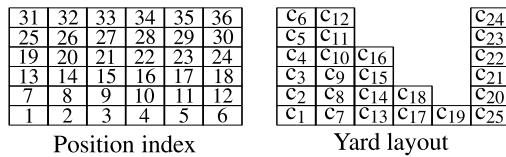


Fig. 2 Example of position index and yard state

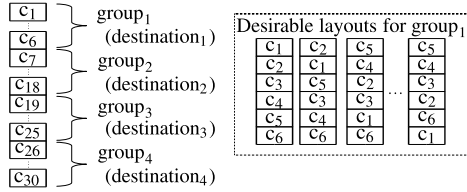


cars from sub-track to sub-track is called removal, and the car-movement from sub-track to main track is called rearrangement. For simplicity, the maximum number of freight cars that each sub-track can have is assumed to be n , the i th car is recognized by a unique symbol c_i ($i = 1, \dots, k$), and the number of sub-tracks is l . Figure 1 shows the outline of freight yard in the case $k = 30, m = n = 6$. In the figure, track T_m denotes the main track, and other tracks A, B, C, D, E, F are sub-tracks. The main track is linked with sub-tracks by a joint track, which is used for moving cars between sub-tracks, or for moving them from a sub-track to the main track. In the figure, freight cars are moved from sub-tracks, and lined in the main track by the descending order, that is, rearrangement starts with c_{30} and finishes with c_1 . When the locomotive L moves a certain car, other cars locating between the locomotive and the car to be moved must be removed to other sub-tracks. This operation is called removal. Then, if $k \leq n \cdot m - (n - 1)$ is satisfied for keeping adequate space to conduct removal process, every car can be rearranged to the main track.

In each sub-track, positions of cars are defined by n rows. Every position has unique position number represented by $m \cdot n$ integers. Figure 2 shows an example of position index for $k = 30, m = n = 6$ and the layout of cars for Fig. 1.

In Fig. 2, the position “aA” that is located at row “a” in the sub-track A has the position number 1, and the position “fF” has the position number 36. For unified representation of layout of car in sub-tracks, cars are placed from the row “a” in

Fig. 3 Example of groups



every track, and newly placed car is jointed with the adjacent freight car. In the figure, in order to rearrange c₂₅, c₂₄, c₂₃, c₂₂, c₂₁, and c₂₀ have to be removed to other sub-tracks. Then, since $k \leq n \cdot m - (n - 1)$ is satisfied, c₂₅ can be moved even when all the other cars are placed in sub-tracks.

In the freight yard, define $x_i (1 \leq x_i \leq n \cdot m, i = 1, \dots, k)$ as the position number of the car c_i , and $s = [x_1, \dots, x_k]$ as the state vector of the sub-tracks. For example, in Fig. 2, the state is represented by $s = [1, 7, 13, 19, 25, 31, 2, 8, 14, 20, 26, 32, 3, 9, 15, 21, 4, 10, 5, 12, 18, 24, 30, 36, 6]$. A trial of the rearrange process starts with the initial layout, rearranging freight cars according to the desirable layout in the main track, and finishes when all the cars are rearranged to the main track.

3 Desired Layout in the Main Track

In the main track, freight cars that have the same destination are placed at the neighboring positions. In this case, removal operations of these cars are not required at the destination regardless of layouts of these cars. In order to consider this feature in the desired layout in the main track, a group is organized by cars that have the same destination, and these cars can be placed at any positions in the group. Then, for each destination, make a corresponding group, and the order of groups lined in the main track is predetermined by destinations. This feature yields several desired layouts in the main track.

Figure 3 depicts examples of desirable layouts of cars and the desired layout of groups in the main track. In the figured, freight cars c₁, ..., c₆ to the destination₁ make group₁, c₇, ..., c₁₈ to the destination₂ make group₂, c₁₉, ..., c₂₅ to the destination₃ make group₃, and c₂₆, ..., c₃₀ to the destination₄ make group₄. Groups_{1,2,3,4} are lined by ascending order in the main track, which make a desirable layout. In the figure, examples of layout in group₁ are in the dashed square.

4 Rearrangement Process

The rearrangement process for cars consists of following 4 operations:

- (1) selection of a freight car to be rearranged into the main track,
- (2) selection of a removal destinations of the cars on the selected car in (1),
- (3) removal of the cars to the selected sub-track,

(4) rearrangement of the selected car.

These operations are repeated until one of desirable layouts is achieved in the main track, and a series of operations from the initial state to the desirable layout is define as a trial.

In the operation (1), each group has the predetermined position in the main track. The car to be rearranged is defined as c_T , and candidates of c_T can be determined by excluding freight cars that have already rearranged to the main track. These candidates must belong to the same group.

Now, define r as the number of groups g_l as the number of freight cars in group l ($1 \leq l \leq r$), and u_{j_1} ($1 \leq j_1 \leq g_l$) as candidates of c_T .

In the operation (2), the removal destination of car located on the car to be rearranged is defined as c_M . Then, defining u_{j_2} ($g_l + 1 \leq j_2 \leq g_l + m - 1$) as candidates of c_M , excluding the sub-track that has the car to be removed, and the number of candidates is $m - 1$.

When rearranging car that has no car to be removed on it is exist, its rearrangement precede any removals. In the case that several cars can be rearranged without a removal, rearrangements are repeated until all the candidates for rearrangement requires at least one removal. If several candidates for rearrangement require no removal, the order of selection is random, because any orders satisfy the desirable layout of groups in the main track. In this case, the arrangement of cars in sub-tracks obtained after rearrangements is unique, so that the movements count of cars has no correlation with rearrangement orders of cars that require no removal.

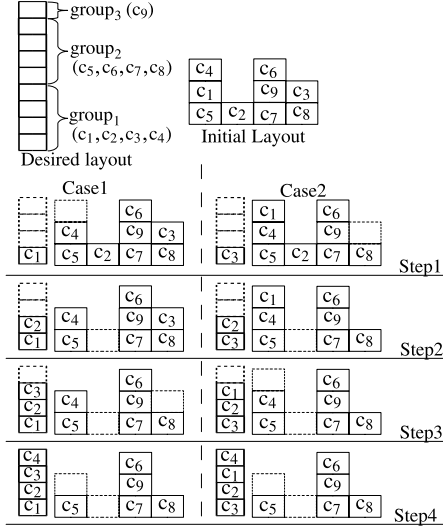
Figure 4 shows an example of arrangement in sub-tracks existing candidates for rearranging cars that require no removal. In the figure, $r = 2$, where c_1, c_2, c_3, c_4 are in group₁, c_5, c_6, c_7, c_8 are in group₂, and group₁ must be rearranged first to the main track. In each group, any layouts of cars can be acceptable. In “Case1” of the example, the rearrangement order of cars that require no removal is c_1, c_2, c_3, c_4 , and in “Case2”, the order is c_3, c_2, c_1, c_4 . Although 2 cases have different orders of rearrangement, the arrangements of cars in sub-tracks and the numbers of movements of cars have no difference.

5 Learning Algorithm

Define $s(t)$ as the state at time t , $s^\dagger(t) = [s(t), c_T]$ and Q_1, Q_2 as evaluation values for $(s(t), u_{j_1})$, $(s^\dagger(t), u_{j_2})$, respectively. $Q_1(s(t), c_T)$ and $Q_2(s^\dagger(t), c_M)$ are updated by following rules:

$$Q_1(s(t), c_T) \leftarrow \begin{cases} \max_{u_{j_2}} Q_2(s^\dagger(t+1), u_{j_2}) & \text{Ⓐ} \\ \text{(next operation is removal),} \\ \max_{u_{j_1}} Q_1(s(t+\tau), u_{j_1}) & \text{Ⓑ} \\ \text{(repetitive rearrangement),} \end{cases} \quad (1)$$

Fig. 4 Direct rearrangements



$$Q_2(s^\dagger(t), c_M) \leftarrow \begin{cases} (1 - \alpha) Q_2(s^\dagger(t), u_{j_2}(t)) \\ \quad + \alpha [R + \gamma \max Q_1(s(t + \tau), u_{j_1}(t + \tau))] \text{ ①} \\ \text{(next operation is rearrangement)} \\ (1 - \alpha) Q_2(s^\dagger(t), u_{j_2}(t)) \\ \quad + \alpha [R + \gamma \max Q_2(s^\dagger(t + 1), u_{j_2}(t + 1))] \text{ ②} \\ \text{(repetitive removal)} \end{cases} \quad (2)$$

where α is the learning rate, γ is the discount factor, τ is the number of direct rearrangements repeated between adjacent selections, and R is the reward that is given when one of desirable layout is achieved.

Propagating Q-values by using Eqs. 1, 2, Q-values are discounted according to the number of removals of cars. In other words, by selecting the movement that has the largest Q-value, the number of removals can be reduced. In the learning stages, each u_j ($1 \leq j \leq g_l + m - 1$) is selected by the following probability:

$$P(s, u_j) = \begin{cases} \frac{\exp(Q_1(s, u_j)/T)}{\sum_{u \in u_{j_1}} \exp(Q_1(s, u)/T)}, & (1 \leq j \leq g_l) \\ \frac{\exp(Q_2(s^\dagger, u_j)/T)}{\sum_{u \in u_{j_2}} \exp(Q_2(s^\dagger, u)/T)}, & (g_l + 1 \leq j \leq g_l + m - 1), \end{cases} \quad (3)$$

where T is a thermo constant.

The proposed learning algorithm can be summarized as follows:

- i Initialize all the Q-values as 0
- ii When no cars are placed on candidates of c_T , all of them are rearranged
- iii If no cars are in sub-tracks, go to **ix** otherwise go to **iv**
- iv ① Determine c_T among the candidates by roulette selection (probabilities are calculated by Eq. 3),
- ② putting reward as $R = 0$,

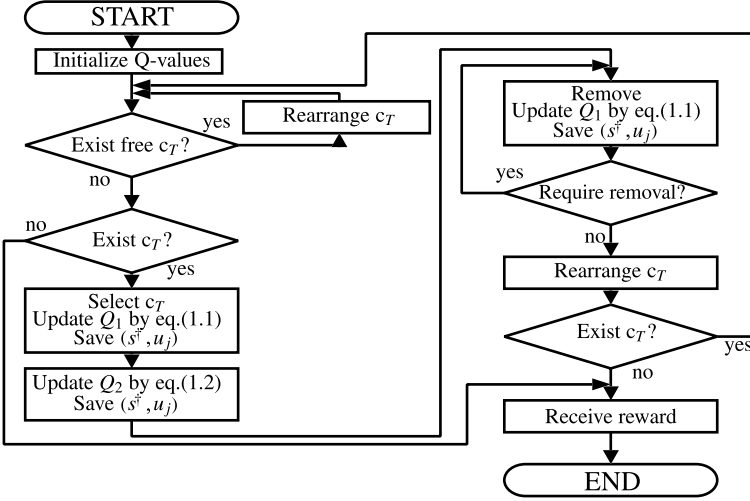


Fig. 5 Flowchart of the learning algorithm

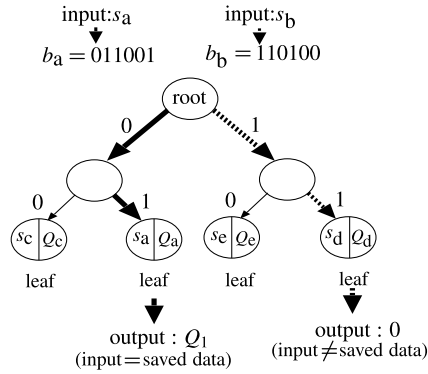
- Ⓒ update the corresponding $Q_1(s, c_T)$ by Eq. 1,
- Ⓓ store s^\dagger
- v If cars to be removed exists, update the corresponding $Q_2(s^\dagger, c_M)$ by Eq. 2a, and store (s^\dagger, c_M) , otherwise update the $Q(s^\dagger, c_M)$ by Eq. 2b and go to ii
- vi Ⓐ If the car to be removed exists, remove it. The destination of the car to be removed is determined by roulette selection (probabilities are calculated by Eq. 3),
 - Ⓑ update the corresponding $Q_1(s, c_T)$ by Eq. 1,
 - Ⓒ store the state s^\dagger ,
 - Ⓓ repeat 6-Ⓐ ~ 6-Ⓒ until all the cars on c_T are removed
- vii Rearrange the c_T
- viii If there exist cars, go to ii
- ix If all the cars are rearranged, the reward R is given, and update a Q-value according to the last movement of car.

Also, flowchart of the proposed learning algorithm is shown in Fig. 5.

6 Data Structure of Look-up Table for Q-value

In the learning algorithm explained previous section, the table lookup method is used for storing and referring Q-values. Since the state of the sub-tracks is represented by $s = [x_1, \dots, x_k]$, $(1 \leq x_i \leq n \cdot m, i = 1, \dots, k)$, the state space requires $(nm)^k$ memory units to store Q-values for each selection. Thus, the number of memory units increases by the exponential rate with increase of the total number of cars k .

Fig. 6 Structure of look-up table



In realistic problems the number of car is often large, so that huge memory size is required in order to store Q-values for all the states. Therefore, in the proposed method, only Q-values corresponding states that have been searched are stored. Binary tree is constructed dynamically during the course of the learning for storing Q-values.

6.1 Specification of a Q-value

In the following, the method to specify a Q-value stored in a look-up table is explained. The input of the table is (s, u_j) , and the output is a Q-value. Assuming I is the order of binary description of $m \cdot n$, the Q-value corresponding to a state s is specified.

$b_i = b_{i1} \cdots b_{iI}$ ($b_{ij} = 0, 1, j = 1, \dots, I$) is defined as the binary description of x_i ($i = 1, \dots, k$). Then, the binary description of s can be described by $b = b_1 \cdots b_k$ of order $(k + 1)I$. That is, a binary tree of depth $(k + 1)I$ can represent s . At each node of the binary tree, by assigning 0 to left descendant of the node and 1 to right descendant, b_{ij} can specify the descendant at the node of depth $I(i - 1) + j$. Each leaf of the tree stores a state and corresponding Q-value. Given an input to the look-up table, the leaf corresponding to the input is specified by a search using b . When the input corresponds to the value stored by the leaf, the look-up table outputs the Q-value stored by the leaf. Otherwise, the table outputs 0. Figure 6 depicts a Q-table constructed by binary tree in the case of $k = m = 2, n = 3, I = 3$. In the figure, inputs $s_a = [1, 3], s_b = [6, 4]$ are given to the look-up table. Since $b_a = [011001]$, in the former case, left, right descendants are specified from the root, the leaf stores the same state as the input s_a , and thus outputs Q_1 . While, in the latter case, since $b_b = [110100]$, right, right descendants are specified, and the leaf stores $s_c (\neq s_b)$. That is, the state that leaf has is different from the input, and thus 0 is output from the look-up table.

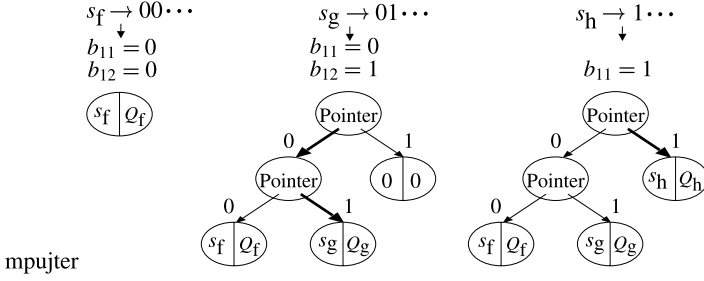


Fig. 7 Construction process of Q-table

Fig. 8 Initial layouts for cases 1, 2

c34	c31	c28	c35	c32	c29														
c1	c4	c7	c13	c16	c10	c19	c22	c25											
c2	c5	c8	c11	c14	c17	c20	c23	c26											
c3	c6	c9	c12	c15	c18	c21	c24	c27	c30	c33	c36								

Case 1

c10	c28	c20	c35	c16	c4	c22	c31	c34	c29	c5	c25								
c2	c6	c8	c11	c14	c17	c36	c23	c26	c21	c32	c19								
c3	c1	c9	c12	c15	c18	c7	c24	c27	c30	c33	c13								

Case 2

6.2 Constitution of Look-up Table

Initially, the tree has only root that has pointer to an empty leaf. When a Q-value corresponding to the input state is different from initial value, the state and the Q-value are stored in the leaf. Then, if a new state that has the same b_{11} as the stored state, 2 consecutive memory units for storing pointer to leafs storing data of state and Q-value are newly allocated. The pointer to the newly allocated memory units is stored in the ascendant node (in this case, root). These operations are repeated until b_{ij} of the new state is different from that of the stored state. If the new state has the different b_{ij} , 1 memory unit is newly allocated for storing the new state and Q-value in addition to 2 memory units for storing pointer to leaf. The pointer to these 2 memory units is stored to the ascendant node. Pointers to leafs are stored in accordance with b_{ij} in 2 memory units that are newly allocated. For nodes that has no descendant nodes or leaf, 0 is stored as the initial value to indicate being empty.

Whenever the state that has updated Q-value appears, the input and the state stored in the specified leaf are compared. When they have the same value, the stored Q-value is update. Otherwise, excluding b_{ij} used to specify the stored state, 2 memory units are newly allocated until first difference between 2 states appears in b_{ij} , and then, a unit is allocated as a leaf to store 2 pairs of state and Q-value.

Figure 7 shows an example of the look-up table for the proposed learning algorithm. In the figure, inputs $s_f = [1, 3]$, $s_g = [2, 1]$, $s_h = [4, 1]$ are given to the table. Then, binary descriptions of s_f, s_g, s_h are $b_f = [b_{11}b_{12}b_{13}b_{21}b_{22}b_{23}] = 001011$, $b_f\beta = 010001$, $b_g = 100001$, respectively. First, s_f is stored in the leaf pointed by the root. Next, when s_g is given, 2 memory units are newly allocated. Since b_{11} of s_f is 0, s_f and Q_α are stored in the left leaf. s_f, s_g have the same b_{11} , and thus, 2 memory units are allocated additionally. Since they have difference in b_{12} , s_f and

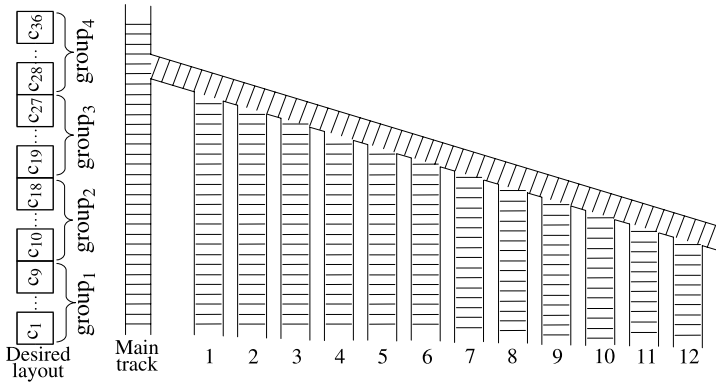


Fig. 9 Yard setting

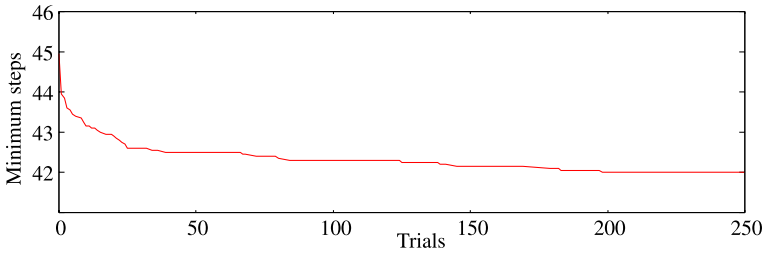


Fig. 10 Minimum steps for case 1

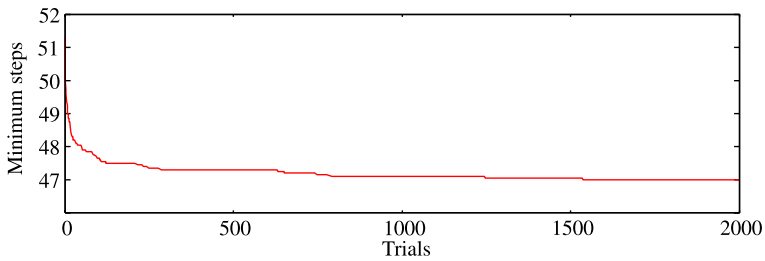


Fig. 11 Minimum steps for case 2

Q_f are stored left leaf, and s_g and Q_g are stored right leaf. Finally, when s_h is given, s_h and Q_h are stored right leaf at the height 1 according $b_{11} = 1$.

The algorithm for look-up table construction is described below.

- i Calculate b from s and initialize $i = j = 1$
- ii If a memory unit corresponding to b_{ij} is a leaf then go to [iii](#), and if it is node then go to [iv](#)

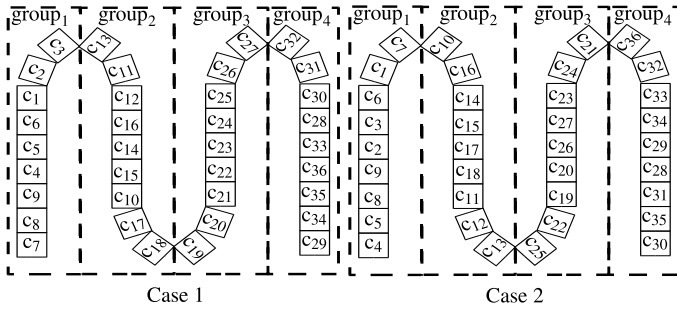


Fig. 12 Final layout

iii update i, j by

$$\begin{cases} j \leftarrow j + 1, i \leftarrow i & (j < I), \\ j \leftarrow 1, i \leftarrow i + 1 & (j = I), \end{cases} \quad (4)$$

and go to ii.

iv Conduct Eq. 4 again, allocate 2 nodes for expanding a tree, and 1 leaf for storing state and Q-value. Then, copy data from original leaf into corresponding leaf, and store the pointers indicating a new leaf and nodes into ascendant nodes.

v If b_{ij} has the same number as the state stored in the leaf, go to iv. Otherwise store the new input and Q-value into the corresponding leaf.

7 Computer Simulations

Computer simulations are conducted for 2 cases. In both cases, $m = 12, n = 6, k = 36$ and initial arrangements of cars in sub-tracks are described in Fig. 8. Also, desirable layout considering groups in the main track is depicted in Fig. 9. In these cases, the rearrangement order of groups is group₁, group₂, group₃, group₄. Cars c_1, \dots, c_9 are in group₁, c_{10}, \dots, c_{18} are in group₂, c_{19}, \dots, c_{27} are in group₃, and c_{28}, \dots, c_{36} are in group₄. Other parameters are set as $\alpha = 0.9, \gamma = 0.9, R = 1.0$.

Figures 10, 11 show results for cases 1, 2. In the figures, horizontal axis expresses the number of trials and the vertical axis expresses the minimum movement counts of cars to achieve a desirable layout found in the past trials. Each result is averaged over 20 independent simulations. In both cases, the movement counts reduce as the number of trials increases. The optimal solution for the case 1 is simple, which can be obtained by removing $c_{28}, c_{29}, c_{31}, c_{32}, c_{34}, c_{35}$ into one of i -th, $10 \leq i \leq 12$ sub tracks. Then, all the cars can be rearranged directly into the main track, and the movement counts of cars is 42. In Fig. 10, within 200 trials, all the simulations derive one of the best solutions that complete a trial by 42 movements of freight cars. For the case 2, several removals are required for some sub-tracks to achieve one of the desired layouts in the main track. In Fig. 11, within 1700 trials, all the simulations derive the solution that completes the trial by 47 movements of freight

cars. In Fig. 12, one of best layouts in the main track obtained by the proposed method is shown for each case. In the figure, positions of cars in the same group are exchanged so that the number of removals required to achieve the layout of groups in the main track is reduced. Thus, in the proposed method, the layout of main track, the rearrangement order of cars, and the removal destination of cars are simultaneously optimized by the autonomous learning.

8 Conclusions

A new scheduling method has been proposed in order to rearrange and line cars in the desirable order onto the main track. The learning algorithm of the proposed method is derived based on the reinforcement learning, considering group of cars. In order to obtain the best solution with feasible memory space, in the proposed method, only referred Q-values are stored in look-up tables constructed dynamically by using binary tree. In computer simulations, by using the proposed method, the layout of main track, the rearrangement order of cars, and the removal destination of cars to achieve the optimal solution has been obtained simultaneously.

References

1. Blasum, U., Bussieck, M.R., Hochstättler, W., Moll, C., Scheel, H.-H., Winter, T.: Scheduling trams in the morning. *Math. Methods Oper. Res.* **49**(1), 137–148 (2000)
2. Dahlhaus, E., Manne, F., Miller, M., Ryan, J.: Algorithms for combinatorial problems related to train marshalling. In: *Proceedings of Australasian Workshop on Combinatorial Algorithms 2000*, pp. 7–16 (2000)
3. He, S., Song, R., Chaudhry, S.S.: Fuzzy dispatching model and genetic algorithms for railyards operations. *Eur. J. Oper. Res.* **124**(2), 307–331 (2000)
4. Hirashima, Y.: A reinforcement learning system for transfer scheduling of freight cars in a train. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19, March 2010*, pp. 112–117 (2010)
5. Jacob, R., Marton, P., Maue, J., Nunkesser, M.: Multistage methods for freight train classification. In: *7th Workshop on Algorithmic Methods and Models for Optimization of Railways*, pp. 158–174 (2007)
6. Kroon, L.G., Lentink, R.M., Schrijver, A.: Shunting of passenger train units: an integrated approach. *Transp. Sci.* **42**, 436–449 (2008)
7. Li, G., Muto, M., Aihara, N., Tsujimura, T.: Environmental load reduction due to modal shift resulting from improvements to railway freight stations. *Q. Rep. RTRI* **48**(4), 207–214 (2007)
8. Tomii, N., Jian, Z.L.: Depot shunting scheduling with combining genetic algorithm and pert. In: *Proceedings of 7th International Conference on Computer Aided Design, Manufacture and Operation in the Railway and Other Advanced Mass Transit Systems*, pp. 437–446 (2000)
9. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Mach. Learn.* **8**, 279–292 (1992)

Coevolving Negotiation Strategies for P-S-Optimizing Agents

Jeonghwan Gwak and Kwang Mong Sim

Abstract In this paper, we consider the negotiation between two competitive agents that consider both time and cost criteria. Therefore, the negotiation agents are designed to not only optimize price utility but also be successful in optimizing (negotiation) speed utility. To this end, the objective of this work is to find effective strategies for the negotiation. The strategies are coevolved through an evolutionary learning process using two different evolutionary algorithms (EAs)—a genetic algorithm (GA) and an estimation of distribution algorithm (EDA). We present an empirical comparison of GA and EDA in coevolving negotiation strategies with different preference criteria in optimizing the price and (negotiation) speed. The experimental results show that both EAs are successful in finding good solutions with respect to both the price-optimizing (*P-Optimizing*) and the speed-optimizing (*S-Optimizing*) negotiation. However, both EAs are not effective in the negotiation for the concurrent optimization of the price and speed (*P-S-Optimizing* negotiation). This is because in some cases, the original fitness function cannot characterize the difference among *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions. Hence, this paper proposes a new fitness function that can better differentiate among the *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions. The experiments showed that the EAs using the proposed fitness function can coevolve effective strategies for the exact *P-S-Optimizing* negotiation.

Keywords Software agent · Price and negotiation speed concurrent optimizing negotiation · Genetic algorithms · Estimation of distribution algorithms

K.M. Sim (✉)

School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, South Korea
e-mail: kmsim@gist.ac.kr

1 Introduction

Automated negotiation is a process in which a group of agents communicate with each other either directly or indirectly to resolve their differences in the hope of eventually reaching an agreement. The interacting agents negotiate to coordinate their activities and to reach mutually acceptable agreements about the division of labor and resources [1].

For a Grid [2] to efficiently support a variety of applications, a resource management system is central to its operation [3]. Grid resource management [3, 4] involves multiple criteria optimization, and some of these criteria are generally classified into time criteria and cost criteria [5]. Sim [6–9] argued and showed that negotiation agents can play an essential role in realizing the Grid vision because the agents can act flexibly for the Grid resources whose performance changes dynamically. To maintain the good performance of the system, the negotiation agents for Grid resource management should be designed to not only optimize price utility (i.e., cost criteria) but be successful in reaching early agreements (i.e., time criteria) [4, 5]. This is because any delay in making a successful negotiation to acquire all the required Grid resources for Grid applications (i.e., resource consumers) before the deadline for executing a job will be considered as an overhead.

Different resource owners and consumers may have different objectives, policies, and preferences [5, 7]. For example, consumers may have conflicting criteria between acquiring cheaper resources and achieving faster negotiation speed (i.e., response time). The resource owner (respectively, a resource consumer) that prefers cheaper resource alternatives at the expense of having to wait longer is said to be more price optimizing (*P-Optimizing*), while a speed-Optimizing (*S-Optimizing*) resource owner (respectively, resource consumer) prefers to obtain a resource more rapidly perhaps by paying a higher price at an earlier round of negotiation than its deadline. Different emphasis on optimizing price and optimizing negotiation speed can be modeled by placing different weights on the two criteria. An exact (or equally distributed) concurrent price and speed optimizing (*P-S-Optimizing*) agent has equal emphasis on the two criteria. In this work, we use three negotiation modes for both competitive agents: *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* negotiation.

The idea of adopting an EDA for coevolving the negotiation strategies of the agents that have different preference criteria such as optimizing price and optimizing negotiation speed was first proposed in [5]. The problem of ineffectiveness in coevolving negotiation strategies for the *P-S-Optimizing* negotiation is presented in [17]. Furthermore, in [17], one possible solution for resolving the difficulties of a *P-S-Optimizing* negotiation is suggested by restricting the solution space (which is the EA's perspective; from the agents' perspective, the negotiation corresponds to the adoption of a feasible strategy space) using predefined strategy profiles. However, in some cases, the fitness function originally used in [5] and [17] cannot effectively discriminate between the different emphases on the two criteria in the total utility space. To solve this problem, we propose a new fitness function that can better characterize the differences among *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions.

The rest of this paper is organized as follows: Section 2 specifies the negotiation model of this work. The coevolutionary framework based on GA and EDA is described in Sect. 3. The problem of coevolving strategies for P-S-Optimizing negotiation and its solution will be presented in Sect. 4. Section 5 presents the experimental results and analyses. Finally, Sect. 6 concludes this work with a summary and suggestions for future work.

2 P-S-Optimizing Negotiation

In a classical bargaining model, the utility function U^x of agent x , where $x \in \{B, S\}$ and \hat{x} denotes x 's opponent, is defined as follows: Let IP_x and RP_x be the initial and the reserve prices of x . Let D be the event in which x fails to reach an agreement with its opponent \hat{x} . $U^x : [IP_x, RP_x] \cup D \rightarrow [0, 1]$ such that $U^x(D) = 0$ and $U^x(P_x) > U^x(D)$ for any possible proposal $P_x \in [IP_x, RP_x]$. If x and \hat{x} are sensitive to time, let τ_x be the deadline of x and $\tau_{\hat{x}}$ be the deadline of \hat{x} . An agreement price that is acceptable to both B and S lies within the interval $[RP_S, RP_B]$.

In the bargaining model with complete information between B and S , both agents know the opponent's initial price, reserve price, and deadline. If one of the agents has a significantly longer deadline than its opponent, the agent that has a longer deadline will have sufficient bargaining advantages. In other words, an agent that has the longer deadline will dominate the negotiation. Under these conditions, Sim [10, 11] proved that an agent's optimal strategy can be computed using its opponent's deadline and reserve price. It can be stated as the following theorem:

Theorem 1 *If agent x 's deadline τ_x is significantly longer than its opponent's deadline $\tau_{\hat{x}}$ (i.e., $\tau_x \gg \tau_{\hat{x}}$), agent x achieves its maximal utility when it adopts the strategy $\lambda_x = \log_{\frac{\tau_x}{\tau_{\hat{x}}}} \left(\frac{RP_{\hat{x}} - IP_x}{RP_x - IP_x} \right)$.*

2.1 Utility Functions

The total (aggregated) utility function U^x of agent x is composed of two attributes—price and time (i.e., the number of negotiation rounds)—and is defined as follows.

$$U^x(P_x, T_x) = w_P \times U_P^x(P_x) + w_S \times U_S^x(T_x) \quad (1)$$

where $P_x \in \{0, P_C\}$ and $T_x \in \{0, T_C\}$, and P_C and T_C is are the price and time at which an agreement is reached. $U_P^x(P_C) \in [0, 1]$ is the price utility of x , and $U_S^x(T_C) \in [0, 1]$ is the speed utility of x . w_P and w_S are the weighting factors for price and (negotiation) speed, respectively, and $w_P + w_S = 1$.

The price and speed utilities are given as follows:

$$U_P^x(P_x) = \begin{cases} u_{\min}^P + (1 - u_{\min}^P) \left(\frac{RP_x - P_C}{RP_x - IP_x} \right), & \text{if an agreement is reached} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$U_S^x(T_x) = \begin{cases} u_{\min}^S + (1 - u_{\min}^S) \left(1 - \frac{T_C}{T_x} \right), & \text{if an agreement is reached} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where u_{\min}^P is the minimum utility when x receives a deal at its reserve price and u_{\min}^S is the minimum utility when x receives a deal at its deadline. For the experimental purpose, the values of u_{\min}^P and u_{\min}^S are defined as 0.1.

If x does not reach an agreement before its deadline, $U_P^x = U_S^x = 0$, and thus, $U^x = 0$. Otherwise, $U^x(P_C, T_C) > 0$.

2.2 Negotiation Strategies

This work considers the bilateral negotiation between B and S with incomplete information in which both agents do not know each other's deadline and reserve price. Both B and S are sensitive to time; further, we adopt the time-dependent strategies proposed in [12]. The proposal P_t^x of x at time t , $0 \leq t \leq \tau_x$ is defined as follows:

$$P_t^x = IP_x + (-1)^\alpha \left(\frac{t}{\tau_x} \right)^{\lambda_x} |RP_x - IP_x| \quad (4)$$

where $\alpha = 0$ for B and $\alpha = 1$ for S , and $0 \leq \lambda_x \leq \infty$.

The time-dependent strategies can be classified into three categories: (i) *conservative* (conceding slowly, $\lambda_x > 1$), (ii) *linear* (conceding linearly, $\lambda_x = 1$), and (iii) *conciliatory* (conceding rapidly, $0 < \lambda_x < 1$) [12, 13].

2.3 Negotiation Protocol

The negotiation between B and S is carried out using Rubinstein's alternating offers protocol [14]. B and S can conduct the negotiation only at discrete time points (e.g., in this work, S makes an offer at $t = 0, 2, 4, 6, \dots$, and B makes a counter-offer at $t = 1, 3, 5, 7, \dots$). The negotiation process ends in both cases: (i) once an offer or a counter-offer is immediately accepted (i.e., an agreement is reached) by the other one or (ii) once the earlier deadline is reached without any agreement. In the latter case, the negotiation process ends in a conflict, and the utility outcome will be zero.

2.4 Objective

For the given different deadlines and different preferences of the cost and time criteria (i.e., different values of w_P and w_S), agents will face different opponents with

different deadlines and strategies. Under these conditions, the objective of this work is to find an effective strategy λ_x that would optimize $U^x(P_x, T_x)$. In this work, (evolutionary) learning is based on two asymmetric populations in which each population has its own fitness evaluation. Agents learn effective strategies by interacting with individuals from the other population through random pairing. In the following section, the detailed procedure will be described.

3 Coevolutionary Models for P-S-Optimizing Negotiation

When populations between two or more species interact, each may evolve in response to the characteristics of the other. The natural coevolution refers to the mutual (or inter-dependent) evolution between interacting populations. The survival skills of the natural coevolution by making mutually beneficial arrangements have long inspired scientists to develop coevolutionary algorithms for highly dependent problems in which there are strong interactions between two elements or among several elements.

In the bilateral negotiation problem domain, inter-population based coevolution having two populations is considered. The fitness of each individual of one population depends on each individual of the other population, and hence, an individual's fitness landscape is not fixed but coupled. Therefore, coevolution is regarded as a type of landscape coupling where adaptive moves by one individual can potentially change the landscape of the other. The interaction between two populations comes from the pairing of strategies of each population, and therefore, a successful pairing mechanism is important. Furthermore, to achieve a better performance, the resulting pairing should make a sufficiently prevailing set in the feasible set. In this work, we use one-to-one random pairing because of its simplicity and efficiency.

To coevolve effective strategies under different deadlines and different weights of price and speed preferences, a coevolutionary framework using real-coded GA and EDA is implemented. B and S have each of their populations d^B and d^S consist of a set of candidate strategies. D^B and D^S are the mating pool (MP) of B and S , respectively.

The evolution of the strategies of one population affects the strategies of the other population. In the process of coevolving strategies, each individual of the two populations will negotiate with the other through one-to-one random pairing. The fitness value of each individual is determined by its negotiation outcome. The details of the GA and EDA are as follows:

3.1 Encoding Scheme

A binary coding mechanism has drawbacks because of the existence of Hamming cliffs and the lack of computation precision [15, 16]. Therefore, in the GA and EDA, each negotiation strategy of the populations is encoded as a real number. Each

Table 1 Characterizations of GA and EDA

EA type	Method	Details
GA [17]	Selection	k -tournament selection
	Crossover	Heuristic crossover
	Mutation	Gaussian mutation
EDA: UMDA [18]	Selection	Truncation selection
	Estimation of probabilistic model	$f_g^B(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X-\mu)^2}{2\sigma^2}}$ where $X = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$, $\mu = \frac{1}{N} \sum_{i=1}^N \lambda_i$, and $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\lambda_i - \mu)^2}$
	Sampling individuals from the model	Sampling normal random variables (individuals) from $f_g^B(X)$ [18, 19]
GA and EDA	Stopping criteria	Terminate when either (i) $g > G^{\max}$ or (ii) $ f_{best}^x - f_{avg}^x < \delta^{stop}$

individual in a population is mainly represented by the strategy that it has. For the experimental purpose, we consider the range of strategies for λ_B and λ_S in $[0, 10]$.

3.2 Fitness Function

To evaluate each individual of a population, the fitness function $f(x)$ is defined as follows:

$$f(x) = U^x(P_x, T_x) = w_P \times U_P^x(P_x) + w_S \times U_S^x(T_x) \quad (5)$$

In each generation g , randomly pick one individual from D_g^B and randomly pick the corresponding individual from D_g^S . Each selection procedure is carried out without a replacement. Then, the selected individuals of one population will negotiate with the selected individuals of the other population in a one-to-one manner. The values of the fitness function will be computed using the resulting negotiation outcomes. Finally, if an agent x reaches an agreement with its opponent, $U_P^x(P_x) > 0$, $U_S^x(T_x) > 0$, and $f(x) = U^x(P_x, T_x) > 0$. If a negotiation is terminated without an agreement, $f(x) = U^x(P_x, T_x) = 0$.

3.3 EAs for Coevolution

The characterizations of GA and EDA used in this work are described in Table 1. For more detailed information of the GA and EDA, refer to [17].

The coevolution procedure for finding the two types of effective strategies (i.e., λ_B and λ_S) from the two populations is described in Algorithm 1. The interaction

between the two populations is in the fitness evaluation stage (at lines 3 and 8 in Algorithm 1).

Algorithm 1 EAs (GA and EDA) for coevolving negotiation strategies

- 1: set the search space $(\lambda_{\min}, \lambda_{\max})$ and the generation counter $g \leftarrow 0$
- 2: generate D_g^x of M randomly generated individuals
- 3: evaluate individuals in D_g^x using the fitness function
- 4: **WHILE** the stopping criteria are not satisfied
- 5: select $N(\leq M)$ individuals d_g^x from D_g^x using the corresponding selection method
- 6: compute the average fitness values of individuals in d_g^x
- 7: **GA:** produce N individuals using the corresponding reproduction mechanism;
 the MP consists of the produced N individuals and d_g^x
- EDA:** produce M individuals using the corresponding reproduction mechanism;
 the MP consists of the produced M individuals
- 8: evaluate individuals in D_g^x using the fitness function
- 9: $g \leftarrow g + 1$
- 10: **ENDWHILE**
- 11: extract the individual with the highest gene value in high fitness value region

4 Problem of P-S-Optimizing Negotiation and Its Solution

4.1 Problem of P-S-Optimizing Negotiation

The GA and EDA described in the previous section cannot coevolve the appropriate strategies for *P-S-Optimizing* negotiation because the original fitness function (which is the same as the utility function described in Sect. 3.2) cannot characterize different emphases (i.e., preference levels) on the cost and time criteria for some cases. That is, for the given values of w_P and w_S , the utility function $U^x(P_x, T_x)$ given in Sect. 3.2 cannot appropriately characterize the difference between $U_P^x(P_x)$ and $U_T^x(T_x)$ in some cases. For example, consider the case of the exact *P-S-Optimizing* negotiation setting $(w_P, w_S) = (0.5, 0.5)$. The total utility $U^x(P_x, T_x)$ of the set $U_P^x(P_x) = 0.7$ and $U_T^x(T_x) = 0.3$ (which corresponds to a more *P-Optimizing* solution) will have the same value for the total utility $U^x(P_x, T_x)$ of the set $U_P^x(P_x) = 0.3$ and $U_T^x(T_x) = 0.7$ (which corresponds to a more *S-Optimizing* solution). Furthermore, it is the same for the set $U_P^x(P_x) = 0.5$ and $U_T^x(T_x) = 0.5$ (which corresponds to the exact *P-S-Optimizing* solution). The above example shows that in some cases, the original fitness function cannot characterize the difference between *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions.

4.2 Proposed Fitness Function

The ambiguity of the original fitness function in the total utility space is solved by adopting the proportion of weighting factors (which represent the preference levels in the total utility space) in the calculation of the proposed fitness function. The key idea of the proposed utility function is to directly measure the difference (or similarity) between (i) the ratio of price and time weighting factors, and (ii) the corresponding ratio of the price and time utility functions. As the difference between them decreases, the value of the fitness function will be considerably high. The proposed fitness function is designed as follows:

$$\text{(Mode 1) for } P\text{-Optimizing: } f(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^x(TC)}{U_P^x(PC)} \right|$$

$$\text{(Mode 2) for } S\text{-Optimizing: } f(x) = 1 - \left| \frac{w_P}{w_S} - \frac{U_P^x(PC)}{U_S^x(TC)} \right|$$

(Mode 3) for *P-S-Optimizing*:

$$f(x) = 1 - \frac{1}{2} \times \left(\left| \frac{w_S}{w_P} - \frac{U_S^x(TC)}{U_P^x(PC)} \right| + \left| \frac{w_P}{w_S} - \frac{U_P^x(PC)}{U_S^x(TC)} \right| \right)$$

5 Empirical Evaluation

A series of experiments was carried out to evaluate the performance of the original fitness function (in Sect. 3.2) and the proposed fitness function (in Sect. 4.2).

5.1 Experimental Settings

An empirical comparison of GA and EDA is also presented to determine which model is more suitable for coevolving negotiation strategies with different preference criteria for optimizing price and speed. The experimental parameter settings are as in Table 2.

For the purpose of the experiments, both competitive agents have the same weights of the preference criteria for the three negotiation modes. The settings for each negotiation mode are as follows: $(w_P, w_S) = (1.0, 0.0)$ for *P-Optimizing* negotiation, $(w_P, w_S) = (0.1, 0.9)$ for *S-Optimizing* negotiation, and $(w_P, w_S) = (0.5, 0.5)$ for *P-S-Optimizing* negotiation.

5.2 Optimal Conditions

In the case of the *P-Optimizing* negotiation, we will experimentally prove the properness by examining two extreme cases: Case I and Case II.

Table 2 Parameter settings for EAs and negotiation

	Parameters	Values
GA and EDA	Population size (N)	25
	Mating pool size (M)	50
	Maximum number of generations (G^{\max})	2500
	Threshold for stopping criteria (δ^{stop})	10^{-4}
	Tournament size (k)	3
	Crossover rate (P_{CX})	0.7
	Mutation rate (P_{MU})	0.002
Negotiation	(IP_B, RP_B)	(5, 85)
	(IP_S, RP_S)	(95, 15)
	$(\lambda_{\min}, \lambda_{\max})$	(0, 10)
	Deadline: Long	100
	Deadline: Mid	50

Case I If an agent B has a sufficient bargaining advantage over S (i.e., $\tau_B \gg \tau_S$), B will dominate the negotiation irrespective of whether S adopts any strategies. The optimal negotiation strategy of B will be determined by Theorem 1. For example, under the negotiation parameter settings described in Table 1, the optimal strategy of B is when $\lambda_B = 3$, and the agreement is reached at $P_C = 15$ and $T_C = 50$. The strategy of S and λ_S will not converge to a specific value, and thus, it will have a dynamic range of values. We describe this dynamic range as [*Min. value*, *Max. value*].

Case II When both B and S do not have any bargaining advantage (i.e., $\tau_B = \tau_S$), we can think that the agreement of the negotiation with the abovementioned negotiation parameters should be reached at a Pareto optimal point (i.e., $P_C = 50$). However, in a practical negotiation model, the point does not always follow the Pareto optimal point since the agent that proposes the first negotiation proposal (in our experiment S does) usually gets a lower payoff (or utility) in the alternating offers protocol. Under the Pareto optimal condition, the optimum strategies are when both λ_B and λ_S equal $\lambda_{\max} = 10$ (i.e., both B and S do not concede at all).

In the case of *S-Optimizing* and *P-S-Optimizing* negotiations, there is no such theory as that in the case of *P-Optimizing* negotiation to prove the optimality of the solutions. However, the evaluation can be carried out by examining whether the solutions follow the general characteristics for the given negotiation mode. For example, in the case of the *S-Optimizing* negotiation, the agreement should be reached at an earlier negotiation time (than its deadline) by paying a relatively high price. The abovementioned two extreme cases are also used for evaluating the optimality of the solutions. Interestingly, according to Proposition 7 given in [13], irrespective of the deadline, agents with linear strategies are more likely to make deals than those

with conservative strategies while achieving higher utility than those with conciliatory strategies. Hence, linear strategies (i.e., $\lambda_B = 1.0$ and $\lambda_S = 1.0$) are optimal solutions for the exact *P-S-optimizing* negotiation.

5.3 Experimental Results

The results of these two extreme cases for the original utility function and the proposed utility function are shown in Tables 3–6.

5.3.1 Results Obtained Using Original Fitness Function

The results are listed in Tables 3 and 4. The original fitness function in Eq. (5) is the same as the total utility function in Eq. (1).

Observation 1 Both GA and EDA with the fitness function in Eq. (5) find good strategies for *P-Optimizing* and *S-Optimizing* negotiation, respectively.

Analysis In both Case I (in Table 3) and Case II (in Table 4), λ_B and λ_S has close to optimum solutions, as discussed in Sect. 5.1.

Observation 2 Both GA and EDA with the fitness function in Eq. (5) cannot find effective candidate solutions for the *P-S-Optimizing* negotiation.

Analysis In both Case I (in Table 3) and Case II (in Table 4), λ_B and λ_S did not converge to the values that we expected (Similarly, by using the EDA proposed in [5], we found that λ_B and λ_S did not converge in the case of *P-S-Optimization*). This is because more conceding strategies will be cut off as the evolution proceeds since these strategies will get a lower payoff than the other strategies (i.e., linear or conservative strategies) eventually. Hence, λ_B and λ_S of both GA and EDA tend to be more *P-Optimizing*.

5.3.2 Results Obtained Using Proposed Fitness Function

The results are listed in Tables 5 and 6. The proposed fitness function does not equal the utility function any longer. The proposed fitness function measures the similarity between the ratio of weighting factors and the ratio of price and speed utilities. For example, for the *P-Optimizing* negotiation (i.e., $(w_P, w_S) = (1.0, 0.0)$), and in Case I (i.e., *B* has significant bargaining advantage than *S*), the optimum utilities are obtained at $(P_C, T_C) = (15, 50)$. For *B*, $U_P^B(P_C) = 0.8875$ and $U_S^B(T_C) = 0.55$. Hence, the optimum value of the proposed fitness function is $f_{opt}^B(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^B(T_C)}{U_P^B(P_C)} \right| = 1 - \left| \frac{0.0}{1.0} - \frac{0.55}{0.8875} \right| = 0.3803$. For *S*, $U_P^B(P_C) = 0.1$

Table 3 Results obtained using original fitness function for Case I

<i>(Long, Mid)</i>	GA	EDA
(w_P, w_S)	(1.0, 0.0)	(1.0, 0.0)
λ_B	2.9574	2.9403
λ_S	[0.1596, 9.8913]	[2.6923, 9.8340]
(P_C^B, T_C^B)	(0.5, 0.5)	(0.5, 0.5)
(P_C^S, T_C^S)	[0.4624, 2.9979]	2.9524
(f_{best}^B, f_{avg}^B)	[0.1012, 9.7045]	[1.3272, 9.4068]
(f_{best}^S, f_{avg}^S)	[(15, 30.5074], [8, 50]]	(15, 50)
N_{Gen}	[(15, 30.5074], [8, 50]]	(15, 50)
	[(15, 30.5074], [8, 50]]	(15, 0003, 50)
	(0.8875, 0.8875)	(0.7188, 0.7188)
	(0.1, 0.1)	(0.1002, 0.1002)
	22.06	15.48
	91.8	15.2
	(0.9377, 0.9378)	(0.9320, 0.9320)
	(0.9332, 0.9332)	(0.9320, 0.9320)
	63.24	44.98
	(0.1, 0.9)	(0.1, 0.9)
	0.1110	0.1178
	0.1674	0.1478
	(51.9763, 1.16)	(49.6131, 1.06)
	(52.3486, 1.16)	(49.8052, 1.06)

Table 4 Results obtained using original fitness function for Case II

(Mid, Mid)	GA	EDA
(w_P, w_S)	(1.0, 0.0)	(1.0, 0.0)
λ_B	9.9931	9.9901
λ_S	9.7487	9.4845
(P_C^B, T_C^B)	(48.1077, 48)	(48.1156, 48)
(P_C^S, T_C^S)	(48.1304, 48)	(48.1310, 48)
(f_{best}^B, f_{avg}^B)	(0.5150, 0.5150)	(0.5149, 0.5149)
(f_{best}^S, f_{avg}^S)	(0.4727, 0.4727)	(0.4727, 0.4727)
N_{Gen}	19.16	24.08
	(0.5, 0.5)	(0.5, 0.5)
	9.5089	8.6693
	9.3096	9.9228
	(47.7413, 46.54)	(50.3378, 47.36)
	(47.8476, 46.52)	(50.3659, 47.36)
	(0.3407, 0.3407)	(0.3187, 0.3187)
	(0.3161, 0.3161)	(0.3226, 0.3227)
	35.4	52.48
	67.1	53.16
	(0.9257, 0.9257)	(0.9297, 0.9298)
	(0.9381, 0.9381)	(0.9367, 0.9368)
	(0.1, 0.9)	(0.1, 0.9)
	0.1162	0.1179
	0.1884	0.1659
	(55.2985, 1.1)	(53.0005, 1)
	(55.8283, 1.1)	(53.1861, 1)

Table 5 Results obtained using proposed fitness function for Case I

<i>(Long, Mid)</i>	GA	EDA
(w_P, w_S)	(1.0, 0.0)	(1.0, 0.0)
λ_B	2.9668	2.9560
λ_S	[0.0177, 9.9634]	[2.0291, 9.2553]
(P_C^B, T_C^B)	(15, 50)	(15.0006, 50)
(P_C^S, T_C^S)	(15.0000, 50)	(14.8506, 49.5)
(f_{best}^B, f_{avg}^B)	(0.3803, 0.3803)	(0.3803, 0.3803)
(f_{best}^S, f_{avg}^S)	(0.0000, 0.0000)	(0.0000, 0.0000)
N_{Gen}	134.1789	43.94
	2059.01	18
	(0.5, 0.5)	(0.1, 0.9)
	0.9738	0.1125
	0.9781	0.0025
	(35.3628, 37.96)	(15.8938, 1)
	(35.4753, 37.01)	(15.7906, 1)
	(0.9986, 0.9999)	(0.2256, 0.2257)
	(0.9832, 0.9894)	(0.9998, 0.9998)
	(0.5, 0.5)	(0.1, 0.9)
	0.9847	0.9847
	0.9584	0.9584
	(35.0521, 37.77)	(35.0521, 37.77)
	(35.0540, 37)	(35.0540, 37)
	(0.9941, 0.9941)	(0.9941, 0.9941)
	(0.9745, 0.9746)	(0.9745, 0.9746)
	59.03	59.03
	(0.1, 0.9)	(0.1, 0.9)
	[3.5351, 9.9581]	[3.5351, 9.9581]
	[0.0400, 9.5237]	[0.0400, 9.5237]
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)
	17.84	17.84

Table 6 Results obtained using proposed fitness function for Case II

(Mid, Mid)	GA	EDA
(w_P, w_S)	(1.0, 0.0)	(1.0, 0.0)
λ_B	10.0000	9.9919
λ_S	9.6512	9.4136
(P_C^B, T_C^B)	(48.0893, 48)	(48.1109, 48)
(P_C^S, T_C^S)	(48.0996, 48)	(48.1432, 48)
(f_{best}^B, f_{avg}^B)	(0.7365, 0.7365)	(0.7359, 0.7359)
(f_{best}^S, f_{avg}^S)	(0.7121, 0.7121)	(0.7123, 0.7124)
N_{Gen}	11.23	17.27
	(0.5, 0.5)	(0.5, 0.5)
	0.9519	0.9508
	1	1
	(50.1952, 28)	(50.1958, 28)
	(50.2000, 28)	(50.2000, 28)
	(0.9909, 0.9910)	(0.9909, 0.9910)
	(1.0000, 1.0000)	(1.0000, 1.0000)
	24.58	19.12
	(0.1, 0.9)	(0.1, 0.9)
	0.1561	0.2595
	0.0026	0.0090
	(15.8203, 1)	(17.4493, 1)
	(15.8099, 1)	(17.4414, 1)
	(0.2167, 0.2167)	(0.2354, 0.2354)
	(1.0000, 1.0000)	(0.9813, 0.9813)
	15.72	41.41

and $U_S^B(T_C) = 0.1$. Hence, the optimum value of the proposed fitness function is $f_{opt}^S(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^x(T_C)}{U_P^x(P_C)} \right| = 1 - \left| \frac{0.0}{1.0} - \frac{0.1}{0.1} \right| = 0.0$. Likewise, for different preference criteria and negotiation parameter settings, different optimum (i.e., maximum) values of the fitness function will be drawn from the proposed fitness function (e.g., for the above example, the maximum value $f_{opt}^B(x)$ for B is 0.3803). More analyses and experiments related to this issue will be presented in our future paper.

Observation 3 Both GA and EDA with the proposed fitness function find effective strategies for the *P-Optimizing* negotiation.

Analysis The results show that both GA and EDA using the new fitness function achieved good performance. In Case I (in Table 5), λ_B converges close to the values of the optimum solutions. Furthermore, in Case II (in Table 6), λ_B and λ_S converge close to the values of the optimum solutions.

Observation 4 In Case I, the GA with the proposed fitness function finds effective candidate solutions for the *S-Optimizing* negotiation. However, the EDA cannot coevolve *S-Optimizing* strategies. In Case II, both GA and EDA can coevolve effective *S-Optimizing* strategies.

Analysis In Case I (in Table 5), the GA can get more conceding strategies that can reach early agreements. However, the EDA cannot find solutions at all. This result shows that the EDA does not have a sufficient search capability to find solutions for the *S-Optimizing* negotiation. In Case II (in Table 6), both GA and EDA can find effective strategies that are considerably more conceding strategies than linear strategies (i.e., $\lambda_B = 0.1561$ and $\lambda_B = 0.0026$ for the GA, and $\lambda_B = 0.2595$ and $\lambda_B = 0.0090$ for the EDA).

Observation 5 In Case I, both GA and EDA with the proposed fitness function find effective candidate solutions for the *P-S-Optimizing* negotiation. The EDA has better performance than the GA in terms of the evolution speed. In Case II, both GA and EDA can coevolve effective *P-S-Optimizing* strategies.

Analysis In Case I (in Table 5), the GA needs considerably more generations to converge to solutions than the EDA (i.e., 2059.01 generations for the GA, and 59.03 generations for the EDA). This result shows that the EDA has better performance with respect to coevolving effective *P-S-Optimizing* strategies in terms of the evolution speed (i.e., the number of generations required for the coevolution). In Case II (in Table 6), both GA and EDA can coevolve effective *P-S-Optimizing* strategies that are close to the value of the linear strategy (i.e., $\lambda_B = 0.9519$ and $\lambda_B = 1$ for the GA, and $\lambda_B = 0.9508$ and $\lambda_B = 1$ for the EDA).

6 Conclusions and Future Work

This paper provides empirical evidence for coevolving negotiation strategies for the negotiation between the two competitive agents having different preference criteria for optimizing price and negotiation speed. Two rather different evolutionary approaches, EDA and GA, were respectively used and compared for coevolving the strategies. The experimental results (given in Tables 3 and 4) showed that adopting the GA and EDA were a good choice for coevolving effective strategies for the *P-Optimizing* and the *S-Optimizing* negotiations. However, we found that both GA and EDA did not converge to proper solutions (that we expected in Sect. 5.1) in the case of the *P-S-Optimizing* negotiation. The main problem of the failure in coevolving the strategies was that the original fitness function (given in [5] and [17]) could not effectively characterize the different weights on the cost and time preference criteria in the total utility space. On the basis of the analysis, we proposed the new utility function in Sect. 4.2. The experimental results (given in Tables 5 and 6) showed that the proposed method achieved more reliable results in the case of the *P-S-Optimizing* negotiation than the method used in [5] and [17]. However, the results also showed that the EDA using the new fitness function did not coevolve effective strategies for the *S-Optimizing* negotiation, and the GA using the new fitness function was not effective in the case of the *P-S-Optimizing* negotiation since it required considerably more generations than EDA. To develop more reliable models that are successful and efficient in all cases, a hybrid model [20] can be adopted to compensate for the defects of each evolutionary approach by combining the GA and the EDA.

Our future work includes an exhaustive analysis that considers (i) more combinations of different preference weights between the cost and the time criteria (e.g., more *P-Optimizing* such as $(w_P, w_S) = (0.7, 0.3)$ and more *S-Optimizing* such as $(w_P, w_S) = (0.3, 0.7)$ cases) and (ii) heterogeneous negotiation (e.g., the case when one agent is *P-Optimizing*, while the other agent is *S-Optimizing*, and all these types of cases). Furthermore, by reducing and focusing on the feasible solution space, the restriction scheme of the solution space in [17] can help to reduce the computational overhead and to obtain more high-quality solutions.

Acknowledgement This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092).

References

1. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge (1994)
2. Foster, I., Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., San Mateo (2003)
3. Krauter, K., Buyya, R., Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. *Softw. Pract. Exp.* **32**, 135–164 (2002)

4. Jarek, N., Jennifer, M.S., Jan, W. Eds.: Multicriteria aspects of grid resource management, in: *Grid Resource Management – State of the Art and Future Trends*, pp. 271–295. Kluwer Academic Publishers, Dordrecht (2003)
5. Sim, K.M.: An evolutionary approach for p-s-optimizing negotiation. In: *Evolutionary Computation, 2008, CEC 2008*, pp. 1364–1371. IEEE World Congress on Computational Intelligence (2008)
6. Sim, K.M.: From market-driven agents to market-oriented grids (position paper). *SIGecom Exch.* **5**, 45–53 (2004)
7. Sim, K.M.: Grid commerce, market-driven G-Negotiation, and Grid Resource Management. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **36**, 1381–1394 (2006)
8. Sim, K.M.: A survey of bargaining models for grid resource allocation. *SIGecom Exch.* **5**, 22–32 (2006)
9. Sim, K.M.: Grid resource negotiation: Survey and new directions. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**, 245–257 (2010)
10. Sim, K.M., Guo, Y., Shi, B.: Adaptive bargaining agents that negotiate optimally and rapidly. In: *IEEE Congress on Evolutionary Computation, 2007, CEC 2007*, pp. 1007–1014 (2007)
11. Sim, K.M., Guo, Y., Shi, B.: BLGAN: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **39**, 198–211 (2009)
12. Sim, K.M., Choi, C.Y.: Agents that react to changing market situations. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **33**, 188–201 (2003)
13. Sim, K.M.: A market driven model for designing negotiation agents. *Comput. Intell.* **18**, 618–637 (2002)
14. Rubinstein, A.: A bargaining model with incomplete information about time preferences. *Econometrica* **53**, 1151–1172 (1985)
15. Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artif. Intell. Rev.* **12**, 265–319 (1998)
16. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: *The Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England (2007)*
17. Gwak, J., Sim, K.M.: Co-evolving best response strategies for p-s-optimizing negotiation using evolutionary algorithms. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong*, pp. 13–18 (2010)
18. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report, EHU-KZAA-IK-4/99, University of the Basque Country (1999)
19. Marsaglia, G., Tsang, W.W.: The ziggurat method for generating random variables. *J. Stat. Softw.* **5**, 1–7 (2000)
20. Peña, J.M., Larrañaga, P., Herves, V., Rosales, F., Pérez, M.S.: GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms, vol. 3029, pp. 1611–3349 (2004)

Policy Gradient Approach for Learning of Soccer Player Agents

Pass Selection of Midfielders

Harukazu Igarashi, Hitoshi Fukuoka,
and Seiji Ishihara

Abstract This research develops a learning method for the pass selection problem of midfielders in RoboCup Soccer Simulation games. A policy gradient method is applied as a learning method to solve this problem because it can easily represent the various heuristics of pass selection in a policy function. We implement the learning function in the midfielders' programs of a well-known team, UvA Trilearn Base 2003. Experimental results show that our method effectively achieves clever pass selection by midfielders in full games. Moreover, in this method's framework, dribbling is learned as a pass technique, in essence to and from the passer itself. It is also shown that the improvement in pass selection by our learning helps to make a team much stronger.

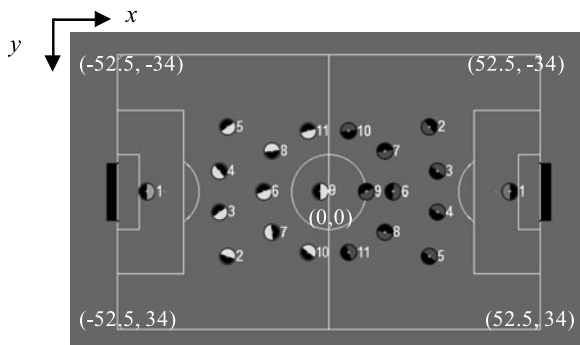
Keywords Multi-agent system · Pass selection · Policy gradient method · Reinforcement learning · RoboCup

1 Introduction

Recently, much work has been done on the learning of coordination in multi-agent systems [1, 2]. The RoboCup Simulation League 2D is recognized as a test bed for such research because there is no need to control real robots and one can focus on learning coordinative behaviors among players. However, multi-agent learning continues to suffer from several difficult problems such as state-space explosion, concurrent learning [3], incomplete perception [4], and credit assignment [2]. These four problems should be studied and resolved to make multi-agent learning successful in the games of the RoboCup Simulation League 2D (Fig. 1).

H. Igarashi (✉)
Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan
e-mail: arashi50@sic.shibaura-it.ac.jp

Fig. 1 RoboCup Soccer Simulation League 2D



As an example of multi-agent learning in a soccer game, Igarashi et al. proposed and applied a reinforcement-learning approach to realizing coordination play between a kicker and a receiver in direct free kicks [5]. They dealt with a learning problem between a kicker and a receiver when a direct free kick is awarded just outside the opponent's penalty area. In such a situation, to which point should the kicker kick the ball? They proposed a function that expresses heuristics to evaluate a candidate target point for effectively sending/receiving a pass and scoring. However, they dealt only with the attacking problems of 2v2 (two attackers and two defenders). In this chapter, we introduce our work in which we applied the method to a pass selection problem of four midfielders in a full soccer game [6].

2 Coordination of Soccer Agents

2.1 Cooperative Play in RoboCup Soccer Simulation

Reinforcement learning is widely used [7, 8] in the research areas of multi-agent learning. In the RoboCup Soccer Simulation League, Andou used Kimura's stochastic gradient ascent (SGA) method to learn the dynamic home positions of 11 players [9]. Riedmiller et al. applied TD learning to learn such individual skills as intercepting the ball, going to a certain position, or kicking and to select those individual skills [10]. They investigated attacking problems with 2v1, 2v2, 3v4, and 7v8. Stone et al. studied keepaway problems [11] with 3v2 and half-field offense problems [12] with 4v5 using Sarsa [7] to learn the selection of macro behaviors such as ball holding, passing, dribbling, and shooting.

2.2 Coordination at Free Kicks

In the previous section, we cited several research efforts on the cooperative behaviors of soccer agents. However, a crucial problem remains. In the previous research,

each agent apparently learns its policy of action selection “autonomously” to complete the given task. Riedmiller et al. assumed that all agents share input information, i.e., the x-y positions of all players and the ball, with other agents [10]. Stone et al. used other agents’ experiences, which are time-series data on state, action, and reward, to accelerate learning in a large problem. For that purpose, agents must communicate their experiences with their partners to facilitate sharing information among them. If agents share input information and experiences with other agents, all agents will obtain the same value function by learning. This will simplify the realization of various cooperative plays among agents. However, if agents’ observations are imperfect or uncertain, none of the agents can share the same input information with all other agents. Without perfect communication among agents, they cannot share their experiences with each other. Moreover, if only agents that have identical value functions are assumed, agent individuality and division of roles among them will not emerge from learning.

Igarashi et al. proposed a method where all agents learn autonomously without assuming perfect communication or identical input information [5]. They applied it to an attacking problem with 2v2 when a direct free kick is awarded just outside the opponent’s penalty area. We briefly summarize the method in the next section.

3 Learning by a Policy Gradient Method

3.1 Policy Gradient Method

A policy gradient method is a kind of reinforcement learning scheme that originated from Williams’s REINFORCE algorithm [13]. The method locally increases and maximizes the expected reward per episode by calculating the derivatives of the expected reward function of the parameters included in a stochastic policy function. This method, which has a firm mathematical basis, is easily applied to many learning problems. One can use it for learning problems even in non-Markov Decision Processes [14, 15]. It was applied to pursuit problems where the policy function consists of state-action rules with weight coefficients that are parameters to be learned [15]. The definition of non-Markov Decision Processes is shown in Appendix 2.

3.2 Stochastic Policy for Action Decision

Igarashi et al. [5] state policy $\pi(a; s, \omega)$ for determining action $a (\in A)$ of all agents when a multi-agent system is in state $s (\in S)$, and this policy is given stochastically by a Boltzmann distribution function with object function $E(a; s, \omega)$ as

$$\pi(a; s, \omega) \equiv \exp(-E(a; s, \omega)/T) / \sum_{x \in A} \exp(-E(x; s, \omega)). \quad (1)$$

Weight parameters $\omega = \{\omega_j\}$ ($j = 1, 2, \dots, N_\omega$) in (1) are determined by a policy gradient method described in the next section. T is a parameter called *temperature*.

3.3 Autonomous Action Decision and Learning

For the autonomous action decisions and the learning of each agent, policy function $\pi(a; s, \omega)$ for the entire multi-agent system was approximated by the product of each agent's policy function [14–16] $\pi_\lambda(a_\lambda; s_\lambda, \{\omega_j^\lambda\})$ as

$$\pi(a; s, \omega) \approx \prod_{\lambda} \pi_\lambda(a_\lambda; s_\lambda, \{\omega_j^\lambda\}), \quad (2)$$

where a_λ is the action of agent λ and s_λ is the state perceived by agent λ . ω_j^λ is the j -th parameter in agent λ 's policy function $\pi_\lambda(a_\lambda; s_\lambda, \{\omega_j^\lambda\})$. In (2), it seems that the correlation among agent action decisions is neglected. However, each agent can see other agents' states, even if they are not perfect, and use them in its policy function $\pi_\lambda(a_\lambda; s, \{\omega_j^\lambda\})$. Thus, the approximation in (2) will partially contribute to learning of coordination among agents.

We assume that agent λ 's policy function $\pi_\lambda(a_\lambda; s, \{\omega_j^\lambda\})$ is also given by a Boltzmann distribution function with objective function $E_\lambda(a_\lambda; s_\lambda, \{\omega_j^\lambda\})$ defined by

$$E_\lambda(a_\lambda; s_\lambda, \{\omega_j^\lambda\}) = - \sum_j \omega_j^\lambda \cdot U_j(a_\lambda; s_\lambda), \quad (3)$$

where function $U_j(a_\lambda; s_\lambda)$ is the j -th heuristics that evaluates action a_λ [5].

At the end of each episode, common reward r is given to all agents that made their decisions during the episode. The derivative of expectation of reward $E[r]$ for parameter ω_j^λ is shown as

$$\partial E[r] / \partial \omega_j^\lambda = E \left[r \sum_{t=0}^{L-1} e_{\omega_j^\lambda}(t) \right], \quad (4)$$

where L is an episode length.

With (2), characteristic eligibility e_ω on the right-hand side of (4) can be written as [14, 15]

$$e_{\omega_j^\lambda}(t) \equiv \partial \ln \pi(a(t); s(t), \{\omega_j^\lambda\}) / \partial \omega_j^\lambda \approx \partial \ln \pi_\lambda(a_\lambda(t); s_\lambda(t), \{\omega_j^\lambda\}) / \partial \omega_j^\lambda. \quad (5)$$

Substituting (2) and (3) into (5), the policy gradient in (4) gives a learning rule as

$$\Delta \omega_j^\lambda = \varepsilon r \sum_{t=0}^{L-1} \left[U_j(a_\lambda(t)) - \sum_{a_\lambda} U_j(a_\lambda) \pi_\lambda(a_\lambda; s_\lambda(t), \{\omega_j^\lambda\}) \right] / T, \quad (6)$$

where $a_\lambda(t)$ is an action actually selected by policy π_λ and $s_\lambda(t)$ is a state perceived by agent λ at time t . Each agent updates ω_j^λ by the learning rule in (6) at the end of each episode [5].

4 Pass Selection Problem

4.1 Pass Selection in Soccer Games

A pass is a typical cooperative play between two players in soccer games. Determining the receiver from among teammates is very important in a pass play. The first action to consider is passing the ball to a receiver safely. However, useless iteration of backward passes should be avoided. Thus, a player must select a receiver who stands in a position to receive the pass safely and who has a relatively high possibility of scoring a goal after receiving the ball. For this purpose, we use heuristics functions that seem to be useful for selecting a pass receiver as $U_j(a_\lambda; s_\lambda)$ in (3), where a_λ is a passer λ 's action of selecting a pass receiver and passing the ball to the receiver.

4.2 Reward

In this paper, we apply the policy gradient method summarized in Sect. 3 to pass selection problems of midfielders. Dribbling is considered a pass from and to the passer itself. If midfielders can pass and dribble the ball safely without being intercepted, the length of time their team holds the ball will be longer. Keeping the ball for as long a time as possible is obviously one of the effective strategies of midfielders in a soccer game. Here, we define the term “team X is keeping the ball” as the situation where the nearest player to the ball is a member of team X and this situation continues for a duration of more than five simulation cycles.

We define learning episode σ of team X by a history of states and actions of agents while team X keeps the ball, and we define reward r of episode σ as $r(\sigma) = -1/L(\sigma)$. Length $L(\sigma)$ of episode σ is defined by the difference between the time when team X begins to keep the ball and the time when the ball is taken by the opponent team. Since r takes a negative real value, it is actually a penalty rather than a reward. The shorter the duration time of keeping the ball is, a larger common penalty $r(\sigma)$ is given to team X 's midfielders who made pass selection during episode σ .

4.3 Heuristics for Pass Selection Problems

We used five heuristics from U_1 to U_5 for evaluating the current position of a teammate as a desirable pass receiver. U_1 , U_2 and U_3 are heuristics for passing the ball safely. U_4 is heuristics for making an aggressive pass. U_5 is used for treating reliable information as more important knowledge than unreliable information. All U_i are normalized between 0 and 10.

The meanings of the five heuristics are summarized as follows. U_1 considers the existence of opponents on the pass course. U_2 evaluates a distance between a receiver and the opponent nearest to the receiver. U_3 takes into account the number of opponent players around a receiver. U_4 expresses a heuristics that the nearer receiver to the opponent's goal has a greater chance of scoring a goal. U_5 evaluates the degree of certainty of a receiver's position perceived by the passer. The definitions of the five heuristics are shown in Appendix 1.

5 Experiments

5.1 Team Used in Learning Experiments

We used UvA Trilearn Base 2003 as a base team for learning experiments. UvA Trilearn 2003 is a team of the University of Amsterdam and champion over 46 qualified teams in the 2D Simulation Soccer League of RoboCup2003 Padova. The team released a part of the source code of UvA Trilearn 2003 [17].

According to the home page of the UvA team [17], the released code contains low-level and intermediate-level implementation (agent-environment synchronization method, world model, player skills) but not a high-level decision procedure. Instead, they have included a simple high-level action selection strategy. The fastest player to the ball intercepts the ball and shoots it to a random corner in the opponent's goal regardless of his position on the field. The remaining players move to a strategic position determined by their home position in the formation and by the position of the ball. We modified the code slightly and implemented the learning function defined in Sects. 3 and 4 for UvA's four midfielders. We call this team "UvA Trilearn Base 2003," or "UvA Base" for short, and use it in our learning experiments.

5.2 Learning Experiments

UvA Base with four midfielders who learn pass selection plays 50 games against UvA Base with midfielders who randomly make passes to teammates. Initial values of ω_j^λ in (3) are set to 0. T and ε in (6) are set to 10.0 and 0.1, respectively.

The change in rewards r given to four midfielders is shown in Fig. 2, where $\langle r \rangle_{50}$ is an average over the last 50 episodes. Midfielders have their own set of ω_j^λ and learn their values by playing 50 games. The results of ω_j^λ are nearly identical to each other. Therefore, only the results of a midfielder called MF_8 are shown in Fig. 3.

Reward averages $\langle r \rangle_{50}$ for all midfielders do not increase in Fig. 2, even as the learning proceeds. At a glance, the learning seems ineffective and a waste of time. However, the effectiveness of learning is shown by evaluation experiments

Fig. 2 Reward average over the last 50 episodes

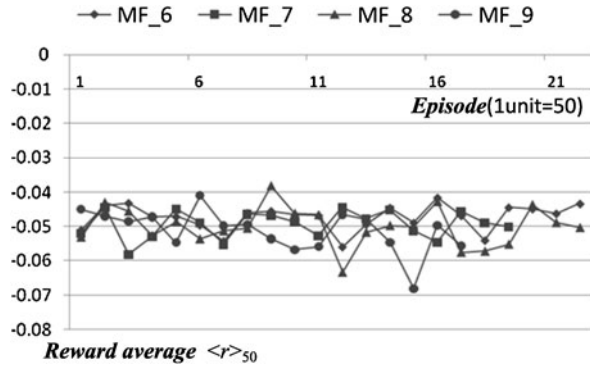
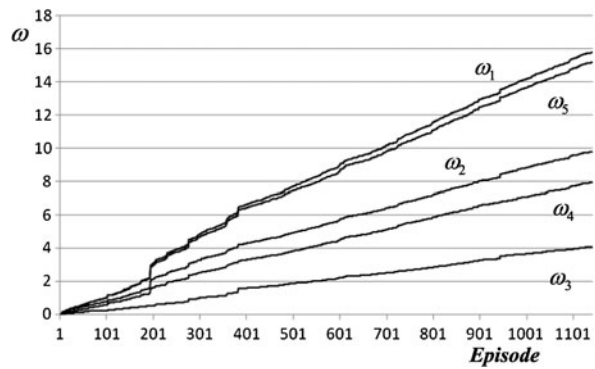


Fig. 3 Change in ω_j of player MF_8



in the next section. The phenomenon of reward averages not increasing is due to the following reason. A player in UvA Base is programmed to keep a close watch over the opponent goal when it possesses the ball. This increases confidence in the information of a teammate’s position in the forward direction of a passer, and it increases the importance of a forward pass in objective function $E(a_\lambda; s, \{\omega_j^\lambda\})$ in (3). Thus, a large value of ω_5 in Fig. 3 causes an agent to make a more offensive pass and to play in a deeper area of the opponent team’s territory as learning proceeds. Playing in a deep area of the opponent prevents the duration of keeping the ball from getting longer.

The ratio among ω_j^λ seems to converge in Fig. 3. This means that there is a strong attractor point in the parameter space of ω_j^λ and that our learning captures this attractor point. Increasing ω_j^λ with a fixed ratio in magnitude is equivalent to annealing in a stochastic policy in (1). Accordingly, a deterministic pass selection algorithm based on ω_j^λ at the attractor point was obtained as policy $\pi_\lambda(a_\lambda; s, \{\omega_j^\lambda\})$ by the learning experiments in this section.

Table 1 Results of 30 games between UvA Base with four midfielders who learned pass selection in N games and UvA Base with midfielders who did not learn but only randomly made passes to teammates

N	Games	Wins-draws-losses	Goals
0	30	14-5-11	31-24
5	30	22-2-6	59-23
10	30	23-2-5	51-17
30	30	23-1-6	46-15
50	30	26-0-4	54-19

5.3 Evaluation Experiments

In order to evaluate the effectiveness of the learning method described in Sects. 3 and 4, we have the team of UvA Base with four midfielders trained in the learning experiments play 30 games against UvA Base with midfielders who did not obtain any learning. The four midfielders in the former team make a deterministic decision because parameter T in their policies is fixed at a very low value in the evaluation experiments. The midfielders in the latter team randomly make passes to teammates because all ω_j^λ are fixed to 0.

Table 1 shows that learning pass selection in only five games ($N = 5$) increases the team's goals and decreases the opponent's goals. Therefore, passes by the four midfielders contribute to both offense and defense. Furthermore, it is occasionally observed in full games that a midfielder selects himself as a receiver and begins dribbling when there is no appropriate receiver to whom the ball can be safely passed. Consequently, they learned how to make a decision on whether they should select passing or dribbling the ball.

6 Conclusion

This research develops a learning method for the pass selection problem of midfielders in RoboCup Soccer Simulation games. A policy gradient method is applied as a learning method to solve this problem because it can easily express the various heuristics of pass selection in a policy function. We implement the learning function in the midfielders' programs of a well-known team, UvA Trilearn Base 2003. Experimental results show that our method effectively achieves clever pass selection by midfielders in full games. Moreover, in this method's framework, dribbling is learned as a pass technique, in essence to and from the passer itself. It is also shown that the improvement in pass selection by our learning helps to make a team much stronger.

In the future, we will apply our learning method to other tasks in a soccer game, such as a receiver's selection of the destination point for receiving a pass. A receiver must move to receive a pass safely, to receive a through pass from a passer, and

to deceive opponent players. Consequently, the receiver must learn some policies and change them depending on the situation. Moreover, an agent becomes a passer and a receiver depending on the time and situation. For example, a passer and a receiver must change their roles at the next moment to succeed with a wall-pass. Accordingly, an agent must change its role and policy at every moment. The next step of our work will focus on how agents learn the most desirable selection of roles and policies in their current situation.

Appendix 1: Heuristics from U_1 to U_5

We define five heuristics from U_1 to U_5 for evaluating the current position of a teammate as a desirable pass receiver. All U_i are normalized between 0 and 10.

(i) $U_1(a_\lambda; s_\lambda)$ considers the existence of opponents on the pass course and is defined by

$$U_1 = \begin{cases} 10.0 & (\text{if } diff > 30 \text{ or } OppDist > AgentDist \text{ or when dribbling}), \\ 9.0 & (\text{if } 20 < diff \leq 30 \ \& \ OppDist < AgentDist), \\ 7.0 & (\text{if } 10 < diff \leq 20 \ \& \ OppDist < AgentDist), \\ 4.0 & (\text{if } diff \leq 10 \ \& \ OppDist < AgentDist), \\ 2.0 & (\text{else}), \end{cases} \quad (7)$$

where $diff$ is the angle between the pass direction and the direction to an opponent. $AgentDist$ is the distance between a passer and a receiver. $OppDist$ is the distance between a passer and the nearest opponent player to the passer (Fig. 4a). If a passer cannot get enough information to calculate all three values, we set $U_1 = 2.0$. If a passer perceives more than one opponent, values of U_1 are calculated for all opponents and the smallest of these values is used.

(ii) $U_2(a_\lambda; s_\lambda)$ evaluates distance min_dist between a receiver and the opponent nearest to the receiver (Fig. 4b). It is defined by

$$U_2 = \begin{cases} 10 & (\text{if } min_dist \geq 30), \\ 10 - (30.0 - min_dist)/5.0 & (\text{if } min_dist < 30), \\ 2.0 & (\text{else}). \end{cases} \quad (8)$$

(iii) $U_3(a_\lambda; s_\lambda)$ takes into account the number of opponent players around a receiver. $OpponentsNum$ is the number of opponent players standing within a 10-meter radius of the receiver:

$$U_3 = \begin{cases} 10 - OpponentsNum & (\text{if } OpponentsNum \leq 10), \\ 0.0 & (\text{else}). \end{cases} \quad (9)$$

(iv) $U_4(a_\lambda; s_\lambda)$ expresses a heuristics that the nearer receiver to the opponent's goal has a greater chance of scoring a goal. It is defined by

$$U_4 = \begin{cases} 10 - goal_dist/10.0 & (\text{if } goal_dist \leq 40.0), \\ 2.0 & (\text{else}), \end{cases} \quad (10)$$

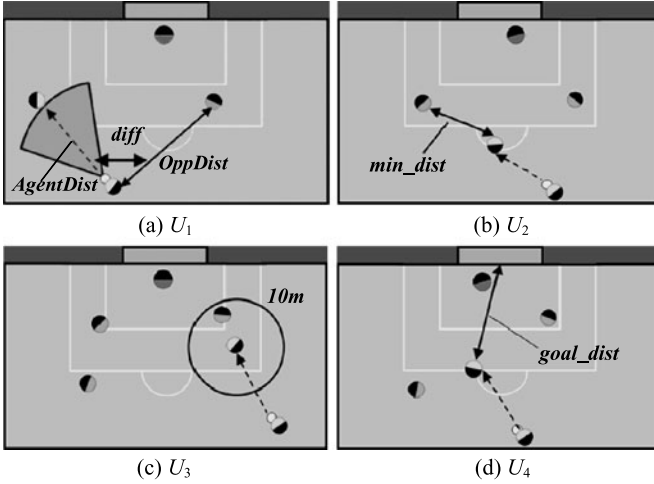


Fig. 4 Heuristics used for a passer to select a receiver

where $goal_dist$ is the distance between a receiver and the goalmouth of the opponent team.

(v) $U_5(a_\lambda; s_\lambda)$ evaluates the degree of certainty of a receiver's position perceived by the passer and is defined by

$$U_5 = \begin{cases} 9.0 & \text{(if a receiver and a passer is the same player),} \\ 10.0 \times confidence & \text{(else),} \end{cases} \quad (11)$$

where $confidence$ is the reliability of the receiver's position. The value $confidence$ is defined by

$$confidence \equiv \max\left(1 - \frac{CurrentCycle - LastSeeCycle}{100}, 0\right), \quad (12)$$

where $CurrentCycle$ is the current time and $LastSeeCycle$ is the latest time when the passer saw a receiver. The later the information is, the higher its reliability is.

Appendix 2: Non-Markov Decision Processes

In Ref. [14], we defined non-MDPs and proved several theorems even in non-MDPs. We show a brief summary of Ref. [14] in this appendix. Non-Markov Decision Processes (*Non-MDPs* or *N-MDPs*) are defined by stochastic decision processes without one or some properties in transition probabilities $P_{ss'}^a = P(s_{t+1} = s' | s_t = s, a_t = a)$, expectation of rewards $R_{ss'}^a = E[r_{t+1} | s_t = s, a_t = a, s_{t+1} = s']$, and agent policy $\pi(s, a) = P(a_t = a | s_t = s)$, where s_t is the state of the environments and a_t is an action of an agent at time t . An agent receives reward r_t at time t . Note that whether a learning problem becomes a problem in MDP or non-MDP

depends on a system designer's definition of s , a , and $\pi(s, a)$. Non-MDPs are processes where at least one of three quantities depends not only on the current state of the environment but also past states or actions. We proved four theorems and two corollaries in Ref. [14]. However, we show only Theorem 1 in this appendix.

MDP is a special case of non-MDP. Theorem 1 shows how the gradient of the expected discounted return in non-MDPs is expressed in MDPs.

Theorem 1 *In MDPs,*

$$\frac{\partial \rho}{\partial \omega_j} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s, a)}{\partial \omega_j} Q^\pi(s, a), \quad (13)$$

where

$$\rho(\pi) \equiv E \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0, \pi \right], \quad (14)$$

$$Q^\pi(s, a) \equiv E \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a, \pi \right], \quad (15)$$

and

$$d^\pi(s) \equiv \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid s_0, \pi). \quad (16)$$

$\rho(\pi)$ is the expected discounted return from initial state s_0 following policy π . $Q^\pi(s, a)$ is an action-value function, and $d^\pi(s)$ is a discounted weighting of states encountered starting at s_0 and then following policy π . The expression in (13) agrees with the result derived independently by Sutton et al. [18] and by Konda and Tsitsiklis [19] for policy gradient learning in MDPs.

References

1. Weiss, G., Sen, S.: *Adaption and Learning in Multi-agent System*. Springer-Verlag, Berlin (1996)
2. Sen, S., Weiss, G.: Learning in multiagent systems. In: Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp. 259–298. The MIT Press, Cambridge (1999)
3. Arai, S., Miyazaki, K.: Learning robust policies for uncertain and stochastic multi-agent domains. In: *7th International Symposium on Artificial Life and Robotics*, pp. 179–182 (2002)
4. Lovejoy, W.S.: A survey of algorithmic methods for partially observed Markov decision processes. *Ann. Oper. Res.* **28**, 47–66 (1991)
5. Igarashi, H., Nakamura, K., Ishihara, S.: Learning of soccer player agents using a policy gradient method: coordination between kicker and receiver during free kicks. In: *2008 International Joint Conference on Neural Networks (IJCNN 2008)*, pp. 46–52 (2008)
6. Igarashi, H., Fukuoka, H., Ishihara, S.: Learning of soccer player agents using a policy gradient method: pass selection. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 31–35 (2010)

7. Sutton, R.S., Barto, A.G.: Reinforcement Learning. The MIT Press, Cambridge (1998)
8. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996)
9. Andou, T.: Refinement of soccer agents' positions using reinforcement learning. In: Kitano, H. (ed.) *RoboCup-97: Robot Soccer World Cup I*, pp. 373–388. Springer-Verlag, Berlin (1998)
10. Riedmiller, M., Gabel, T.: On experiences in a complex and competitive gaming domain—reinforcement learning meets RoboCup. In: *The 2007 IEEE Symposium on Computational Intelligence and Games (CIG2007)*, pp. 17–23 (2007)
11. Stone, P., Kuhlmann, G., Taylor, M.E., Liu, Y.: Keepaway soccer: from machine learning test bed to benchmark. In: Bredenfled, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) *RoboCup 2005: Robot Soccer World Cup IX*, pp. 93–105. Springer-Verlag, New York (2006)
12. Kalyanakrishnan, S., Liu, Y., Stone, P.: Half field offense in RoboCup soccer – A multiagent reinforcement learning case study. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) *RoboCup-2006: Robot Soccer World Cup X*, pp. 72–85. Springer-Verlag, Berlin (2007)
13. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992)
14. Igarashi, H., Ishihara, S., Kimura, M.: Reinforcement learning in non-Markov decision processes – statistical properties of characteristic eligibility. *IEICE Trans. Inform. Syst.* **J90-D(9)**, 2271–2280 (2007) (in Japanese). This paper is translated into English and included in *The Research Reports of Shibaura Institute of Technology, Nat. Sci. Eng.* **52(2)**, 1–7 (2008). ISSN 0386-3115
15. Ishihara, S., Igarashi, H.: Applying the policy gradient method to behavior learning in multiagent systems: the pursuit problem. *Syst. Comput. Jpn.* **37(10)**, 101–109 (2006)
16. Peshkin, L., Kim, K.E., Meuleau, N., Kaelbling, L.P.: Learning to cooperate via policy search. In: *16th Conference on Uncertainty in Artificial Intelligence (UAI2000)*, pp. 489–496 (2000)
17. UvA Trilearn 2003: <http://staff.science.uva.nl/~jellekok/robocup/2003/>
18. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: *Advances in Neural Information Processing Systems 12 (NIPS'99)*, pp. 1057–1063 (2000)
19. Conda, V.R., Tsitsiklis, J.N.: Actor-critic algorithms. In: *Advances in Neural Information Processing Systems 12 (NIPS'99)*, pp. 1008–1014 (2000)

Genetic Algorithm for Forming Buyer Coalition with Bundles of Items in E-Marketplaces

Anon Sukstrienwong

Abstract The benefits of buyer coalitions are well-known for electronic marketplaces. However, a few existing buyer coalition schemes over the Internet have focused on forming a buyer coalition with bundles of items. This paper presents an algorithm to form a buyer coalition with bundles of items by using genetic algorithms (GAs). The algorithm called GAGroupBuying finds the best disjoint subsets of all buyers based on the total utility which addresses the situation where a whole group of buyers can be partitioned into smaller sub-groups to obtain more utility than they could accomplish in the whole group. The proposed algorithm is compared with a previous algorithm called GroupPackageString as shown by Boongasame and Sukstrienwong (Emerging Intelligent Computing Technology and Applications, pp. 674–685, 2009). The results of GAGroupBuying simulation are found to be satisfactory with the total discount of a buyer coalition.

Keywords Genetic algorithm · Buyer coalition · Bundles of items · Coalition structure

1 Introduction

To date, sellers prefer to put their products on the electronic marketplaces because it is a big channel to sell their products in a large number. And, several commercial websites such as <http://buy.yahoo.com.tw>, <http://www.amazon.com>, and <http://www.staples.com> usually offer a volume discount for customers if a number of selling is big. In common, buyers prefer to obtain a deduction from the price list offered by sellers in return for payment. One common shopping tactic which most buyers are likely to make is a group buying because a large group of buyers

A. Sukstrienwong (✉)

Information Technology Department, School of Science and Technology, Bangkok University,
Bangkok, Thailand

e-mail: anon.su@bu.ac.th

gains more negotiating power. So, they can advantageously bargain with sellers to get a great discount on their purchases. Most buyer coalition schemes form a buyer coalition in e-marketplaces to gain the discount for buying a large number of goods as shown by Chang et al. [2], He and Ioerger [4], and Chen et al. [5]. Then, the Combinatorial Coalition Formation scheme of Li and Sycara [3] considers an e-market where each buyer places a bid on a combination of items with reservation prices, and sellers offer price discounts for each item based on volume. Also, the work of Ito et al. [6] presented an agent-mediated electronic market by group buying scheme. Buyers or sellers can sequentially enter a market to make their decisions. Tsvetovat et al. [7] have investigated the use of incentives to create buying group. Yamamoto and Sycara, [8] presented the GroupBuyAuction scheme for forming buyer coalition based on item categories. Then, the paper of Hyodo et al. [9] presented an optimal coalition formation among buyer agents based on a genetic algorithm (GA) with the purpose of distributing buyers among group buying site optimally to get good utilities. These strategies address the situation where different buyers participate in one group to purchase goods at low cost. So, a whole group of buyers can advantageously deal with sellers to gain more discount for a large volume of items. However, most researchers do not consider forming a group buying with bundles of items which addresses the situation where a whole group can be partitioned into smaller sub-groups to obtain more utility than they could accomplish in the whole group.

In the corresponding conference paper Sukstrienwong [13] to this paper, further results are found. The proposed algorithm called GAGroupBuying is applied to the set of buyers who want to purchase some particular items within several packages. The concentration of this paper is to find the best formation of a group buying which can give the better group's utility. GAs are a heuristic search scheme based on a model of Darwinian evolution. So, there is no guarantee for convergence of GAs, but the experiments have shown that the approach is acceptable. The paper is divided into five sections, including this introduction section. Section 2 outlines the problem of buyer coalition with bundles of items. The detail of the GAGroupBuying is shown in Sect. 3. Experimental results of the GAGroupBuying and discussions are provided in Sect. 4. Finally, conclusions and further work are provided in Sect. 5.

2 Outlines the Group Buying with Bundles of Items

In this paper, there are some assumptions applying to the group buying with bundles of items. First, the group buying is formed under one goal to maximize a buyer utility which can be calculated by the discount received by being a member of a coalition. Also, the definition of bundles of items is a slightly difference from the work of Gürle et al. [10]. In this paper, a bundle of items refers to several items together in a package of one or more goods at one price. Additionally, buyers have several choices of items, and they have seen the price list of all packages provided by sellers before they can place their orders. And, if the package is pure bundling,

Table 1 An example of price lists

Sellers	Package numbers	Product types				Price (\$)
		Toilet paper	Paper tower	Lotion	Detergent	
s ₀	0	pack of 1	–	–	–	12.00
	1	pack of 4	–	–	–	36.08
	2	–	pack of 4	–	–	40.21
	3	–	–	pack of 1	–	7.89
	4	–	–	pack of 3	pack of 2	18.93
s ₁	0	–	–	–	pack of 9	47.21
	1	–	–	–	pack of 1	8.00
	2	–	pack of 4	pack of 3	–	40.64
	3	pack of 2	pack of 4	–	–	59.76
	4	–	pack of 2	pack of 3	pack of 3	52.00

Table 2 An example of buyer's orders with the reservation prices

Buyers	Buyer's order (number of items × (reservation prices \$))			
	Toilet paper	Paper tower	Lotion	Detergent
b ₀	2 × (9.1)	–	1 × (6.55)	–
b ₁	–	–	–	2 × (5.95)
b ₂	–	–	3 × (6.0)	1 × (6.0)
b ₃	1 × (9.1)	4 × (10.0)	–	–

the average price of each item will be cheaper than the price of the same product in a single item package. An example that is given to describe a motivation for this paper is described below.

There exist several commercial websites providing some attractive products with the special prices. Suppose there are four types of products selling in an e-marketplace which are toilet paper, paper tower, lotion, and detergent. Sellers make a price list for their products; both single item and bundles of items, as shown in Table 1. For instance, package number 3 of seller 1, denoted P_1^3 , comprises of two packs of toilet paper and four packs of paper tower. This package is sold at the special price of \$59.76. However, a single pack of toilet paper and a four pack of paper tower are sold separately at the price of \$12.00 and \$40.21 by seller 0, as seen in package number 1 and 3 of seller 0, respectively. A symbol '–' appearing in Table 1 means that sellers do not put a specific product in that package number.

Suppose there are some buyers requesting to purchase some products listed in Table 1. Due to buyer demands and economic problems, some buyers do not want to purchase a whole package by their own, so they come to participate in the group buying with the aim of obtaining their requested products at a good price on the purchasing. Suppose there are four buyers joining into a particular group. Each buyer, denoted as b_m where $0 \leq m < 4$, places some orders to specific items with reser-

vation prices as shown in Table 2. For instance, a buyer b_0 wants to get 2 packs of toilet paper at a reservation price of \$9.10 of each and a pack of lotion at a reservation price of \$6.55. The reservation of each means the price for particular item that buyers are willing to pay. Also, a symbol ‘—’ in Table 2 under each product means that a buyer has no request to buy. If a buyer called b_0 goes directly to purchase those products by his own, the total cost that b_0 needs to pay for two packages of P_0 and one package of P_3 from seller s_0 is $12.0 * 2 + 7.89 = 31.83$ dollars, while b_0 is willing to pay totally for those items at $2 * 9.1 + 6.55 = 24.75$ dollars. It seems to be impossible for b_0 to purchase all items along. So, b_0 would come to participate in a group with the aim of obtaining better prices on the purchasing.

3 Genetic Algorithms for Buyer Coalition with Bundles of Items

3.1 The Basic Concept of GA

Genetic algorithms (GAs) are a heuristic search algorithm based on the evolutionary ideas of natural selection. The basic mechanism of GAs is designed to simulate processes in nature necessary for evolution. GAs often apply to find the optimal solution to the problem by manipulating a population of solutions. The manners of problems need to be encoded in chromosomes for distinguishing good solutions from bad ones. Only good individuals of the population survive to the next generation while bad individuals are eliminated from the selection process. The genetic algorithm begins with generation 0 with the completely random population size M . There exist two operations to perform during one generation, crossover and mutation. During the run, the given individual might be mutated and crossed within single generation. Note that the GAs work on fixed-length character strings. The simplest algorithm represents each chromosome as a bit string which are strings of 1's and 0's. In general, GAs search the space of possible character strings in an attempt to find good string based on fitness value of strings.

There are few steps in preparing to use the genetic algorithm of fixed-length character strings to solve a problem as shown by Koza [11], genetic programming on the programming of computers by means of natural selection.

1. Selecting a representation that facilitates the solution of the problem by means of the genetic algorithm.
2. The fitness value of each element, which could be the objective of solution, is used to distinct good and bad individuals from the population. In general, the fitness measure is inherent in the character string it encounters.
3. For setting up the genetic algorithm, primary parameters for controlling the genetic algorithm are the population size (M) and the maximum number of generations to be run (Gen). Secondary parameters, the crossover probability (p_c), and the mutation probability (p_m) are required to create new population.

4. Randomly create an initial population of individual fixed-length character strings. Evaluate the fitness value of each individual in the population. Create a new population for the next generation based on fitness value by randomly applying two operators, crossover operator and mutation operator which are described below.
 - **Crossover operation:** Create two new offspring from two existing members in the population by recombine randomly chosen substring from selected members.
 - **Mutation operation:** Create a new offspring from an existing member in the population by randomly mutating the character at one position in the fixed-length string.

3.2 Problem Formalization

Let $B = \{b_0, b_1, b_2, \dots, b_n\}$ denote the collection of buyers. Each buyer wants to buy several items in the e-marketplaces (Figs. 1, 4, 7). There is a set of seller $S = \{s_0, s_1, s_2, \dots, s_j\}$. One seller called s_i has made special offers within a set of packages, denoted as $P_i = \{P_i^0, P_i^1, P_i^2, \dots, P_i^k\}$, where k is the maximum number of packages. A price per item is a monotonically decreasing function when the size of the package is increasing big, and each package is associated with the set of prices, denoted $\text{Price}_i = \{\text{price}_i^0, \text{price}_i^1, \text{price}_i^2, \dots, \text{price}_i^k\}$. Each package P_i^k is comprised of multiple items associated by a vector $\{g^{i,k}_0, g^{i,k}_1, g^{i,k}_2, \dots, g^{i,k}_j\}$. If a goods named $g^{i,k}_j$ is not available in the package, then $g^{i,k}_j = 0$. Each b_n needs to buy some items offered by sellers, denoted as $Q_n = \{q^n_0, q^n_1, q^n_2, \dots, q^n_j\}$, where q^n_j refers to the quantity of items g_j of buyer n . If $q^n_j = 0$, it means that b_n does not have a request to purchase g_j . Also, b_n places the reservation price for each particular goods associated with Q_n , denoted as $R_{s_n} = \{rs^n_0, rs^n_1, rs^n_2, \dots, rs^n_j\}$ where $rs^n_h \geq 0, 1 \leq h \leq j$. As stated earlier, there is a situation where a whole group of buyers can be partitioned into smaller groups to receive a better utility than they could get by joining everyone in the whole group. A group of buyers can be divided into small groups of buyers called sub-coalitions, denoted as C_n which $C_n \subseteq B$ and $C_0 \cap C_1 \cap \dots \cap C_n = \emptyset$. There is a situation where a coalition structure CS is a partition of B of which each buyer of B belongs to exactly one coalition and some buyers may be alone in their coalitions. For example, if $B = \{b_0, b_1, b_2, b_3\}$, there are 15 possible coalitions: $\{b_0\}, \{b_1\}, \{b_2\}, \{b_3\}, \{b_0, b_1\}, \{b_0, b_2\}, \{b_0, b_3\}, \{b_1, b_2\}, \{b_1, b_3\}, \{b_2, b_3\}, \{b_0, b_1, b_2\}, \{b_0, b_1, b_3\}, \{b_0, b_2, b_3\}, \{b_1, b_2, b_3\}$, and $\{b_0, b_1, b_2, b_3\}$. From the work of Sandholm et al. [12], there are $2^n - 1$ possible coalition structures for n buyers. This is simply the number of subsets of B (not counting the empty set). So, for buyers = 4 there are $2^4 - 1 = 16 - 1 = 15$ possible coalition structures: $\{\{b_0\}, \{b_1\}, \{b_2\}, \{b_3\}\}, \{\{b_0\}, \{b_1\}, \{b_2, b_3\}\}, \{\{b_2\}, \{b_3\}, \{b_0, b_1\}\}, \{\{b_0\}, \{b_2\}, \{b_1, b_3\}\}, \{\{b_1\}, \{b_3\}, \{b_0, b_2\}\}, \{\{b_0\}, \{b_3\}, \{b_1, b_2\}\}, \{\{b_1\}, \{b_2\}, \{b_0, b_3\}\}, \{\{b_0\}, \{b_1, b_2, b_3\}\}, \{\{b_0, b_1\}, \{b_2, b_3\}\}, \{\{b_1\}, \{b_0, b_2, b_3\}\}, \{\{b_0, b_2\}, \{b_1, b_3\}\}, \{\{b_2\}, \{b_0, b_1, b_3\}\}, \{\{b_0, b_3\}, \{b_1, b_2\}\}, \{\{b_3\}, \{b_0, b_1, b_2\}\},$ and $\{b_0, b_1, b_2, b_3\}$.

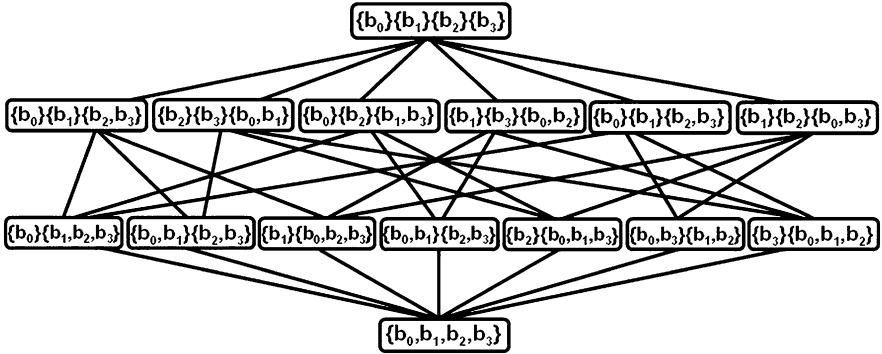


Fig. 1 A coalition structure graph of 4 buyers

Fig. 2 An example of BuyerChromosome of $\{\{b_0, b_1\}, \{b_2, b_3\}\}$ $\{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$

3.3 Problem Representation for Forming Buyer Coalition

In this paper, the BuyerChromosomes are designed to facilitate genetic algorithm processes. A BuyerChromosome is the chromosome of sub coalition together with a sequence of random packages. For instance, suppose a set of buyer $B = \{b_0, b_1, b_2, b_3\}$ is randomly divided into two sub-groups, $C_0 = \{b_0, b_1\}$ and $C_1 = \{b_2, b_3\}$. And, the chosen packages for sub-coalitions are arranged in the form of $\langle P^i_j \rangle^L$, where L is the maximum number of packages. An index i refers to the number of sellers, and an index j means the number of the packages provided by seller s_i . Suppose the algorithm has randomly chosen $\langle P^2_1 P^2_1 P^4_0 \rangle$ for C_0 and $\langle P^3_1 P^2_0 P^4_1 \rangle$ for C_1 . Then, a BuyerChromosome of $\{\{b_0, b_1\}, \{b_2, b_3\}\}$ is built as shown in Fig. 2.

From Fig. 2, C_0 buys two packages of P_2 from seller s_0 and one package of P_4 from seller s_0 . And, C_1 buys one package of P_3 and P_4 from seller s_1 and one package of P_2 from seller s_0 . In common, if there exist n buyers who want to buy products from i sellers offered to sell k different packages, then the possible number of BuyerChromosomes is $(2^m - m - 1)(i)(k^L)$, where L is the maximum number of packages of each BuyerChromosome.

3.3.1 Fitness Measurement

In this paper, the fitness value to each BuyerChromosome in the population is the total utility of a coalition. The utility of a coalition is a discount obtained from sellers by purchasing goods. The sum of all buyers' reservations of the sub-coalition denoted as $reservation(C_n)$ is shown as follows:

$$reservation(C_n) = \sum_{b_m \in C_n} \sum_{j=1}^k r s_j^m * q_j^m. \tag{1}$$

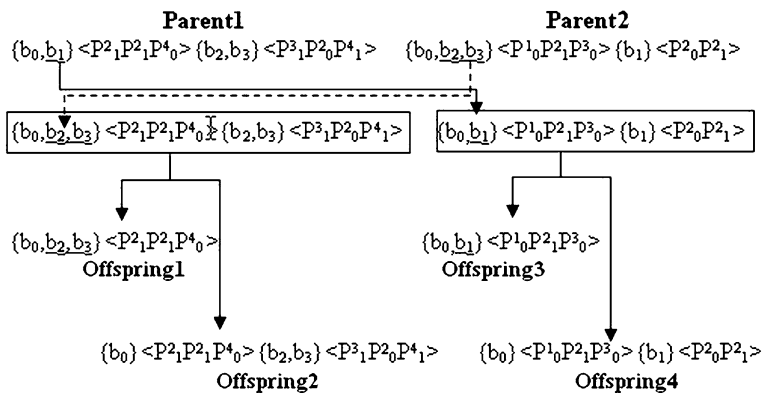


Fig. 3 An example of buyer crossover of C_0 at buyer-crossover point = 2

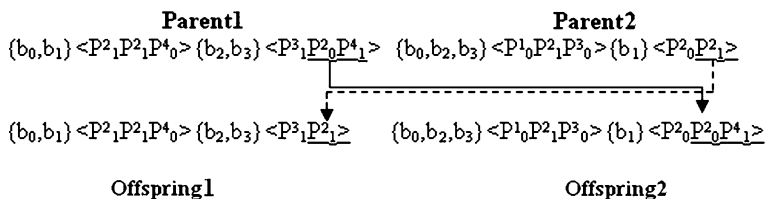


Fig. 4 An example of package crossover of C_1 at package-crossover point = 2

And the total amount paid by a sub-coalition, called C_n , to sellers denoted as $AmountPaid(C_n)$ is

$$AmountPaid(C_n) = \sum_{j=0}^L Price_j^i. \quad (2)$$

The total utility of C_n , denoted as $Utility(C_n)$, is the defined as follows:

$$Utility(C_n) = reservation(C_n) - AmountPaid(C_n). \quad (3)$$

So, the fitness function of each individual in the population of the coalition is mathematically defined as follows:

$$Fitness_function = \sum_n Utility(C_n). \quad (4)$$

3.3.2 Breeding a New Population of BuyerChromosomes

For each new solution to be produced, a BuyerChromosome is selected for breeding based on its fitness value. By producing new offspring, in the GAGroupBuying methods of crossover and mutation are described below.

1. **Crossover operation:** This operator creates new offspring from existing parents of current generation in the population by having crossover operation at a random crossover point. Suppose two selected BuyerChromosomes in crossover with $L = 3$ are $\{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$ and $\{b_0, b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_1\} \langle P^2_0 P^2_1 \rangle$. The parents will be split at the crossover point into two substrings, a crossover fragment and a remainder. In this paper, there are three types of crossover operation which are buyer crossover, package crossover, and sub-coalition crossover.

- Buyer crossover: Suppose the GAGroupBuying randomly selects C_0 with a buyer-crossover point = 2 for both parent1 and parent2 shown in Fig. 3. Two possible resulting of recombining the crossover fragment and the remainder of two substrings are $\{b_0, b_2, b_3\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$ and $\{b_0, b_1\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_1\} \langle P^2_0 P^2_1 \rangle$. As the result shown, these new BuyerChromosomes are not completely done because each new BuyerChromosome has the problem where $C_0 \cap C_1 \neq \emptyset$. So, these new BuyerChromosomes are required to split into new BuyerChromosomes. Finally, four new BuyerChromosomes are generated which are $\{b_0, b_2, b_3\} \langle P^2_1 P^2_1 P^4_0 \rangle$, $\{b_0\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$, $\{b_0, b_1\} \langle P^1_0 P^2_1 P^3_0 \rangle$, and $\{b_0\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_1\} \langle P^2_0 P^2_1 \rangle$.
- Package crossover: If the GAGroupBuying chooses to operate with the package crossover, the sub-coalition of current BuyerChromosome and a package-crossover point is randomly selected. Suppose the GAGroupBuying randomly select C_1 with the buyer-crossover point = 2 for both parent1 and parent2. Two possible resulting are $\{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_1 \rangle$ and $\{b_0, b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_1\} \langle P^2_0 P^2_0 P^4_1 \rangle$.
- Sub-coalition crossover: This operator chooses the sub-coalition of each parent and replaces it into the other parent. Suppose C_0 of parent1 and C_1 of parent2 are randomly selected. Then C_0 of parent1 is replaced into the position of C_1 of parent2. Also, C_1 of parent2 is replaced into the position of C_0 of parent1. This process generates two new strings, $\{b_1\} \langle P^2_0 P^2_1 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$ and $\{b_0, b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle$. However, the second string requires to be regulated because b_0 is allocated in two sub-coalitions. So, $\{b_0, b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle$ is split into new strings which are $\{b_0, b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle$ and $\{b_2, b_3\} \langle P^1_0 P^2_1 P^3_0 \rangle \{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle$. The result of the sub-coalition crossover is shown in Fig. 5.

2. **Mutation operation:** This operator creates a new offspring from best existing BuyerChromosomes in the population. The mutation process begins by randomly selecting a sub-coalition and choosing a number as a mutation point. The chromosome at this mutation point is randomly changed. Suppose the random selecting BuyerChromosome is $\{b_0, b_1\} \langle P^2_1 P^2_1 P^4_0 \rangle \{b_2, b_3\} \langle P^3_1 P^2_0 P^4_1 \rangle$. And, a sub-coalition C_0 with mutation point = 1 are chosen in random. Then, the possible resulting of mutation operation is shown in Fig. 6.

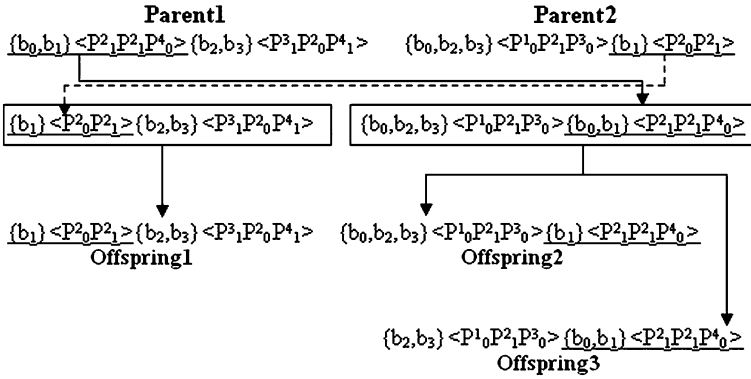
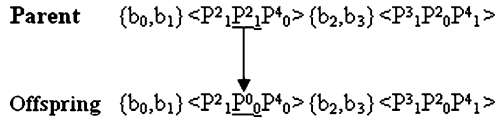


Fig. 5 An example of sub-coalition crossover

Fig. 6 An example of mutation operator



The GAGroupBuying keeps generating a new population for each generation until it gets the maximum number of generations to be run. At last, the best Buyer-Chromosome, which is possible to yield the highest profit to the group buying based on the fitness value might be found. Suppose the best BuyerChromosome of $B = \{b_0, b_1, b_2, b_3\}$ is $\{b_0\} <P^1_0 P^2_1 P^3_0> \{b_2, b_3\} <P^3_1 P^2_0 P^4_1>$. Each sub-coalition has its utility which is calculated by Eq. 4. This buyer coalition is formed without buyer b_1 , so the total utility of the coalition is calculated by Eq. 5.

$$Total_Utility = \sum_{N \subset \{\{b_0\}, \{b_2, b_3\}\}} Utility(C_N) \tag{5}$$

4 Experimental Results and Discussions

The experimental results of the proposed algorithm are derived from a simulation which has implemented more than 4000 lines of C++ program. It is run on a Pentium(R) D CPU 2.80 GHz, 2 GB of RAM, IBM PC. The primary parameters for controlling the proposed algorithm, as seen in Table 3, are the population size (M), the maximum number of generations to be run (Gen), the crossover probability (p_c), and the mutation probability (p_m). The other parameters are for setting sellers and buyers used in the simulation. There are two sellers ($NumOfSeller = 2$) which are arbitrarily set to sell several products in packages, while there are ten buyers ($NumOfBuyer = 10$) requesting to buy several items with the reservation prices. The maximum number of items a package is 5 ($MaxItemPerPackage = 5$). Some packages listed by sellers are pure bundling packages, but some packages are single-item

Table 3 The summarized parameters for the GAGroupBuying simulation

Constants	Detail	Range
M	Population size	1000
p_c	Crossover probability	0.95
p_m	Mutation probability	0.05
NumOfSeller	No. of sellers	2
NumOfBuyer	No. of buyers	10
MaxItemPerPackage	Max number of items per package	5
MaxNumOfPackagePerSeller	Max number of packages for each seller	10

Table 4 A price list of two sellers

Sellers	Package numbers	Product types				Prices (\$)
		A	B	C	D	
s_0	0	1	-	-	-	1000
	1	2	1	-	-	2690
	2	4	-	-	-	3400
	3	-	-	1	-	900
	4	-	-	4	1	4200
s_1	0	-	1	1	-	1700
	1	-	-	1	1	1740
	2	-	1	-	-	900
	3	-	4	-	-	3580
	4	-	1	-	1	1970
	5	-	-	-	1	955
	6	-	-	-	5	4250
	7	1	-	1	1	2750
8	1	1	-	1	2700	

packages. All price lists offered by two sellers is shown in Table 4. And, there are ten buyers who want to participate in the group buying. An example of buyers' orders with reservation prices are listed in Table 5.

The experimental result received from the simulation is shown in Table 6. It shows that the GAGroupBuying finds the better total utility than the GroupPackageString. The group buying can be formed by separating buyers into two sub-coalitions, $\{b_0, b_1, b_2, b_3, b_4, b_6, b_8\}$ and $\{b_5, b_7, b_9\}$. The best of BuyerChromosomes found by the GAGroupBuying during the simulation is $\{b_0, b_1, b_2, b_3, b_4, b_6, b_8\} < P^2_0 > \{b_5, b_7, b_9\} < P^1_1 P^4_0 >$ with the total utility of \$100. The number of buyers who can purchase the items is 9. And, the average total utility earned from GAGroupBuying is 89.12. A buyer who fails to join with any sub-coalitions is b_0 , so it is removed out of the group buying. The result is also shown in graphs

Table 5 An example of ten buyers' orders

Buyers	Buyer's order (number of items \times reservation price \$)			
	A	B	C	D
b_0	–	$1 \times (860)$	–	–
b_1	–	–	–	$1 \times (860)$
b_2	–	–	$2 \times (855)$	–
b_3	–	–	$1 \times (860)$	–
b_4	–	–	–	$1 \times (855)$
b_5	$1 \times (860)$	–	–	–
b_6	–	–	$1 \times (865)$	–
b_7	$1 \times (860)$	–	–	–
b_8	–	–	$1 \times (860)$	–
b_9	$2 \times (855)$	–	–	–

Table 6 Experimental results

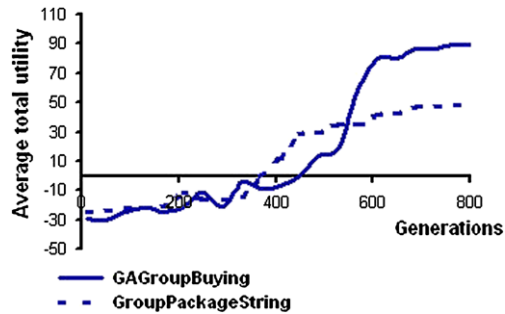
GAGroupBuying	GroupPackageString			
Number of sub-coalitions of (the best result)	Total buyers who can purchase the items	Avg. of total utility (10 runs)	Total buyers who can purchase the items	Avg. of total utility (10 runs)
$\{\{b_0, b_1, b_2, b_3, b_4, b_6, b_8\}, \{b_5, b_7, b_9\}\}$	9	89.12	10	48.14

the comparison result of GroupPackageString and GroupPackageString. Brief justifications for the values of this graph are $M = 1000$, $p_c = 0.95$, and $p_m = 0.05$. At the beginning of the graph, when the number of generations is low, both algorithms perform almost similar. However, when the number of generations is about 550, the curve of the GAGroupBuying graph increases rapidly over the cure of the GroupPackageString graph. It is because the quality of the GAGroupBuying for forming a buyer coalition improves. One prominent characteristic designed for the GAGroupBuying is that buyers with bad reservation be able to identify, and they possible to be removed out of the group buying. So, the average total utility earned by the GAGroupBuying is higher than the average total utility earned by GroupPackageString.

5 Conclusions and Future Work

The paper presents an algorithm called GAGroupBuying for forming a buyer coalition with bundle of items. The aim of this algorithm is to maximize the total utility

Fig. 7 The comparison result of GAGroupBuying and GroupPackageString



of the group buying. The algorithm based on GA works by randomly partitioning a whole group of buyers into smaller sub-groups with the aim of obtaining more utility (or discount) than they could receive in the whole group. From the experimental result, the GAGroupBuying is able to find better utility than the GroupPackageString, but the GAGroupBuying requires a bigger number of generations to be run. And, there are some restrictive assumptions for this algorithm. The buyer coalition is formed concerning only the price attribute, and buyers of the coalition cannot buy the bundles of items by themselves. All buyers do not know each others in the coalition. Sellers in electronic marketplaces can supply unlimited items of any products. If the package is pure bundling, the average price of each item will be cheaper than the price of a single item package. These restrictions can be extended in future researches.

References

1. Boongasame, L., Sukstrienwong, A.: Buyer coalitions with bundles of items by using genetic algorithm. In: *Emerging Intelligent Computing Technology and Applications*, pp. 674–685. Springer, Heidelberg (2009)
2. Chang, Y., Li, C., Smith, J.R.: Searching dynamically bundled goods with pairwise relations. In: *Proceedings of ACM Electronic Commerce*, pp. 135–143 (2003)
3. Li, C., Sycara, K.: Algorithm for combinatorial coalition formation and payoff diversion in an electronic marketplace. In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 120–127 (2002)
4. He, L., Ioerger, T.: Combining bundle search with buyer coalition formation in Electronic Markets: A distributed approach through explicit negotiation. *Electron. Commerce Res. Appl.* **4**(4), 329–344 (2005)
5. Chen, D., Jeng, B., Lee, W., Chuang, C.: An agent-based model for consumer-to-business electronic commerce. *Expert Syst. Appl.* **34**(1), 469–481 (2008)
6. Ito, T., Hiroyuki, O., Toramatsu, S.: A group buy protocol based on coalition formation for agent-mediated e-commerce. *IJCIS* **3**(1), 11–20 (2002)
7. Tsvetovat, M., Sycara, K.P., Chen, Y., Ying, J.: Customer coalitions in electronic markets. In: Dignum, F., Cortés, U. (eds.) *AMEC III. LNAI*, vol. 2003, pp. 121–138 (2001)
8. Yamamoto, J., Sycara, K.: A stable and efficient buyer coalition formation scheme for e-marketplaces. In: *Proceedings of the 5th International Conference on Autonomous Agents*, Montreal, Quebec, Canada, pp. 576–583 (2001)

9. Hyodo, M., Matsuo, T., Ito, T.: An optimal coalition formation among buyer agents based on a genetic algorithm. In: 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE'03), Laughborough, UK, pp. 759–767 (2003)
10. Gürle, U., Öztop, S., Şen, A.: Optimal bundle formation and pricing of two products with limited stock. *Int. J. Product. Econ.* **118**(2), 442–462 (2009)
11. Koza, J.: *Genetic Programming on the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge (1992)
12. Sandholm, T.W., Larson, K.S., Andersson, M.R., Shehory, O., Tohme, F.: Coalition structure generation with worst case guarantees. *Artif. Intell.* **111**, 209–238 (1999)
13. Sukstienwong, A.: Buyer formation with bundle of items in e-marketplaces by genetic algorithm. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, Hong Kong, 17–19 March 2010, pp. 158–162 (2010)

Inside Virtual CIM

Multi-agent Based Resource Integration for Small to Medium Sized Manufacturing Enterprises

Ning Zhou, Sev Naglingam, Ke Xing,
and Grier Lin

Abstract Worldwide cooperation among manufacturing companies is increasingly gaining importance to face emerging challenges in manufacturing. The traditional Computer Integrated Manufacturing (CIM) systems cannot satisfy the needs of global market as they are deployed only within an enterprise. Therefore, a more flexible and comprehensive integrating methodology is required to overcome distance barriers, facility sharing problems and communication obstacles. These issues lead to the concept of Virtual CIM (VCIM). In this paper, the limitations of current agent based implementation of VCIM concept are analyzed. It also describes approaches to address those limitations and propose further development on Agent based Resource Scheduling Process in Small and Medium Enterprises in VCIM network.

Keywords Virtual CIM · VCIM · Small and medium manufacturing enterprises · Multi-criteria selection · Agent based systems

1 Introduction

Manufacturing enterprises face challenges from worldwide competitors and struggle to extend their business globally in today's economic climate. Therefore, these enterprises tend to use fully integrated manufacturing systems so that they can have the capabilities to rapidly respond to constantly changing customer requirements and produce high quality products in shortest possible time with lowest possible cost. However, Small and Medium Manufacturing Enterprises (SMMEs) are having difficulties in achieving this flexibility and competing with large companies, since SMMEs often do not have enough resources. By integrating manufacturing

N. Zhou (✉)

AME, University of South Australia, Mawson Lakes Boulevard, Mawson Lakes, South Australia 5095, Australia

e-mail: Ning.Zhou@postgrads.unisa.edu.au

resources of many partner enterprises (which may be located at different regions), these enterprises can form a globally integrated SMME network and achieve a competitive edge [1].

Virtual Computer Integrated Manufacturing (VCIM) is a new concept for Computer Integrated Manufacturing (CIM). In 1997, it was proposed by Lin in his keynote in Singapore. He stated the VCIM concept would be the future evolvement of CIM [1]. The main improvement for the word “Virtual” is to address the limitation of CIM definition. In the traditional concept, CIM is often limited within an enterprise. However due to the global competition and collaboration, a solution is needed to expand CIM to a much wider border. Thus Virtual CIM (VCIM) was defined as a network of interconnected global CIM systems, extends traditional concept of CIM from a local and centralized company to world-wide cooperation [2]. Virtual Computer Integrated Manufacturing (VCIM), which is a network of interconnected global Computer Integrated Manufacturing (CIM) systems, extends traditional concept of CIM from a local and centralized system to worldwide cooperation [2]. Today, Companies are expanding their business boundaries locally and internationally, merging or co-operating with others across geographical demarcation. VCIM aims to unite and integrate all activities in an enterprise or a network of enterprises to share resources and management objectives through information integration, in a cohesive manner to work as a seamless global CIM system. In a VCIM system, manufacturing resources may belong to different enterprises or be located at different areas, but all have intention of working together in an integrated manner. After receiving a product order, the VCIM system schedules and organize distributed resources as a temporary production system based on working status information of the resources in real time. As this temporary system disappears when purpose are fulfilled, compared to the tight connection of manufacturing resources in traditional CIM systems, this transient system with VCIM is in a virtual status. To describe this status viewpoint, the word Virtual is used in the concept of VCIM (Virtual CIM).

The effort for global collaboration in manufacturing industry never ends. VCIM aims to integrate all manufacturing resources from different entities in different location, and make them work together as a seamless global CIM system [3]. When receiving a customer order, this system searches among available resources and schedules a best path for production schedule. Those resources in the schedule will form a temporary system to fulfill that customer order. The “virtual” comes from the concept of Virtual Enterprise (VE) studies [4–6]. VE is defined as a collection of companies that are dynamically composed when needed and dissolved after its goal is completed [7]. VE intends to integrate all elements from raw materials in a supply chain as a whole to provide to final customers [8]. However, VCIM has a much wider scope. It has many correlations with other integration concepts such as Intelligent Manufacturing System [9], Holonic Manufacturing System [10]. The aim of the VCIM is to establish cohesive connections among those manufacturing resources. Those resources are often in the form of Small to Medium Manufacturing Enterprises (SMMEs). VCIM is expected to give those SMMEs enhanced competitiveness in Global competition [11].

This article is the revised/extended work based on the paper presented in 2010 IMECS conference [12] and is divided by two parts. The first part reviews the background of VCIM research project. The second part analyzes the current development with Agent-based approaches to build the proposed VCIM system, including the limitations in current system design and potential issues for the system implementation. In the second part, possible solutions to above limitations are discussed. Major changes to current architecture will be demonstrated.

2 Current Development and Limitations

SMMEs play a significant role in pioneering new technologies, markets, and creation of knowledge based industries, all of which are important for future growth and jobs of many countries. In addition, they are often characterized by niche specialist markets in which they have expertise. They succeed by providing high levels of responsiveness and personalized service. In many instances, they can offer lower priced products compared to other large enterprise, because SMMEs have less overheads and their labour and management force are the same. However, when SMMEs grow and develop a typical functional structure, they need to develop efficiencies within their total processes to remain competitive. Many growing SMMEs see their profit margin drop as these inefficiencies mount.

Global competition and today's open markets are driving the enterprises to introduce high quality products and services economically and efficiently. A VCIM system is a strategic move that requires manufacturing enterprises to establish close relationships in order to exploit each other's core competencies for the betterment of the SMME network.

Participation in a VCIM is especially challenging for SMMEs. Since, the VCIM activities involve complex operations and these participants of a VCIM system are distributed worldwide, the SMMEs must overcome global boundaries in terms of distances, time, regulatory constraints, as well as cultural and political differences, for reaching mutually-beneficial agreements on how to optimize the customer order fulfillment process. In addition, some of these partnerships are dynamic and becoming virtual representing the transient status of the collaboration [3].

2.1 Agent Based System Architecture

To optimize resource sharing and to provide a dynamic integration, an agent-based VCIM architecture has been developed. In the agent based VCIM system, three categories of agents have been identified [4]. These agents include Facilitator Agents, Customer Agents, and Resource Agents. Facilitator Agents are designed to act as coordinators to route the information flow across the VCIM agent community. Customer Agents are designed to provide interfaces for customer to participate in the

VCIM system. Finally, Resource Agents are designed as agent interface to encapsulate distributed manufacturing functional entities and connect them with the agent community. The functionalities and responsibilities are described in earlier VCIM research [1, 3–5].

In VCIM agent-based architecture, all the agents are connected to the Internet. Facilitator Agents need to register to Customer Agents and Resource Agents need to register to Facilitator Agents. All the communication is delivered via the Internet [1].

After receiving the customer order, the Customer Agent passes the order to a Facilitator Agent. The Facilitator Agent then works together with those Resource Agents registered to it and make an optimized production schedule [1].

The optimized production schedule is defined as the cheapest schedule with shortest duration time while it satisfies the customer's required due date/time and delivery destination [3, 4].

By connecting a Facilitator Agent, a Customer Agent and many Resource Agents through the Internet, these agents can form a basic multi-agent VCIM system. Nevertheless, a real VCIM system includes many Facilitator Agents, Customer Agents and Resource Agent while the functionalities of Resource Agents may include design, manufacture, delivery, material supply and others [5].

2.2 *Current Limitation*

Current Agent-based VCIM architecture still need improvement, because we have found three major limitations that prevent the Agent-based VCIM architecture into real practice. Those are:

1. Network limitation

The VCIM agents reside across the boundaries of many enterprises. According to past research [6], the agent communication between distributed locations is often unreliable. Thus when performing a resource scheduling across the VCIM network, the big number of VCIM agents, extensive volume of exchanging messages, limited bandwidth and unreliable nature of the Internet will slow down the whole process. Unlike a faster and more reliable Intranet inside a single organization, the VCIM system must use the Internet more wisely with less communication volumes and more flexible mechanisms.

2. Agent selection limitation

The VCIM network is designed to be dynamic. This means any time a new VCIM agent may join or quit the network. Current architecture does not address the mechanism how a Customer Agent finds suitable Facilitator Agents and selects the most suitable one among them. For a particular customer order, the most suitable Facilitator Agent means the Facilitator Agent who most satisfies the Customer Agent with its offer content and other comparable factors.

3. Multiple criteria selection limitation

When making selection from proposed work schedule or manufacturing resources, current architecture only uses two factors: cost and time [5]. While in real situation, many other factors need to be considered, such as: quality, friendship, credit, and delivery reliability [7–11]. Lack of a systematical multi-criteria selection method will limit us reaching the optimal result.

2.3 Research Focus

The research focus is to address the above limitations and give possible solutions to enhance the functionality of the Multi-agent based VCIM. Therefore the focus can be divided into three parts.

1. For network limitation, messages exchange over the network must be minimized. More efficient mechanisms need to be used to optimize message flow and lower network overhead and latency. This part will mainly focus on a redesign of current agent network architecture.

2. For agent selection limitation, the way for agent communication needs to be improved. This part will focus on changes to agent negotiation and agent communication protocol. Detailed steps from customer order to final delivery need to be specified.

3. For multiple criteria selection limitation, multi-criteria needs to be implemented in two selection processes. First process is for a Customer Agent to find and select the best suitable Facilitator Agent for a particular order. Second process is for a Facilitator Agent to find and select the best suitable Resource Agent. This part will focus on solving the limitation in these two selection processes.

3 Agent-Based Architecture and Resource Scheduling Process

This research explores on the revolution of VCIM resource scheduling process and there are three major changes in system architecture, communication protocol and decision making process.

To improve the performance and functionality, an artificial intelligent decision making process is needed to be integrated to the Multi-agent framework. A multi-criteria approach is also needed to be integrated into the agent searching and selection process. According to the research focus addressed before, the new improvements over the new VCIM Agent Architecture and Framework based on three parts:

1. Registry Service
2. VCIM Agent Communication Process
3. Multi-criteria selection Integration

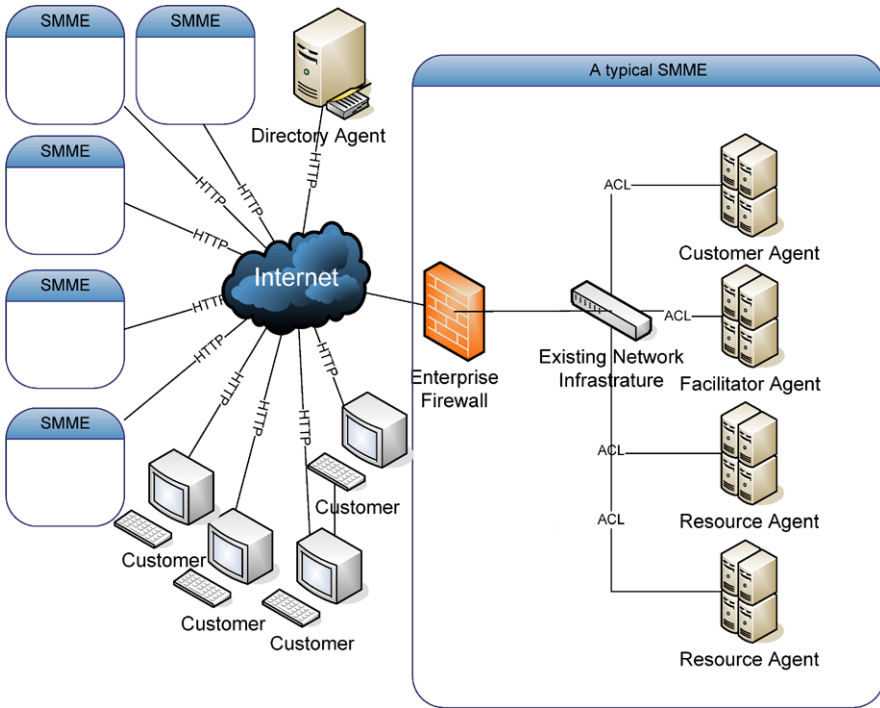


Fig. 1 New VCIM architecture

Figure 1 shows the new designed VCIM architecture. All small to medium sized manufacturing enterprises are connected to the Internet, forming a virtual collaboration network. The VCIM Agents reside in each SMME. Customers use their computer to submit orders to the Customer Agent through the Internet. The new Public Directory also connects directly to the Internet. It stands alone, not within any SMME. And it holds all basic information of each VCIM agent and allows queries for agents. The agents inside a SMME will communicate in ACL (Agent Communication Language). The agents in different SMMEs will communicate in HTTP/SOAP in order to pass enterprise firewalls.

3.1 Registry Service

In order to improve the agent communication over Internet, a central registry like database that holds all the information of each agent across the whole interconnected VCIM network needs to be created to ease the multi-criteria agent searching. Here we define it as a Registry Service. A special agent called Directory Agent is created to provide this service.

Figure 2 shows the improvement to current connection structure. We can see that along with other agents, the new Directory agent is connected to the Internet. They

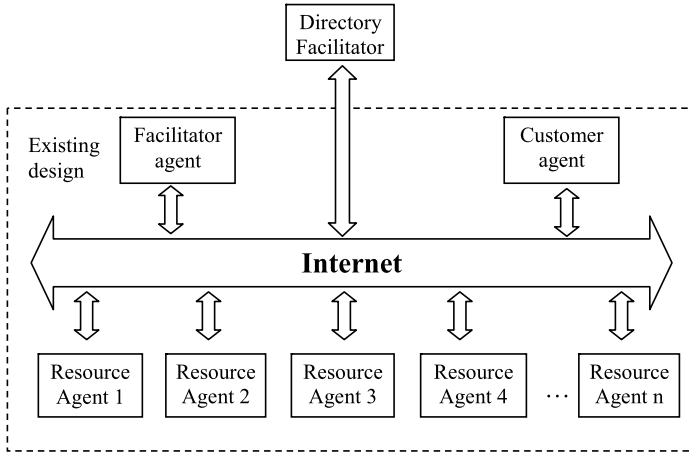


Fig. 2 Connection structure of a base VCIM system (adapted and modified [5])

virtually link together to form a Virtual CIM (VCIM) network. The VCIM network is dynamic because any time a VCIM agent may join or quit. When a new agent joins in the VCIM network, it needs to register itself to the Directory. When a new agent quits in the VCIM network, it needs to de-register itself from the Directory. When an agent wants to search for another agent, it submits criteria-based queries to the Directory and gets result from the Directory. In old implementation, when a new agent joins in the VCIM network, it needs to register itself to all Facilitator Agents. When a facilitator wants to find suitable Resource Agents, it needs to make queries to all Resource Agents. The advantage of a directory service is obvious. Not only can the communication volume be minimized, but also better search response time.

Figure 3 shows the improvement in current information flow. The messages are transported among different types of agents. All agents need to communicate with the new Directory Agent for functions like Registering, De-registering, and Search. The figure also shows that Customer Agent needs talk to Facilitator Agent and Facilitator Agent needs talk to Resource Agents. The communication volume among them is greatly reduced, because the Directory can pre-select suitable agents for them to talk to. In old architecture, the Customer Agent sends requests to all Facilitator Agents. In new architecture, the Facilitator Agent finds suitable Resource Agents that can provide the parts and only send requests to them.

3.2 Agents Based Resource Planning

When a Facilitator Agent receives a request from a Customer Agent, it generates all possible production schedules. Then it finds the best suitable production schedule.

Table 1 shows the each step when a customer order comes in. When a Facilitator Agent receives a request from a Customer Agent, it divides the order into subtasks

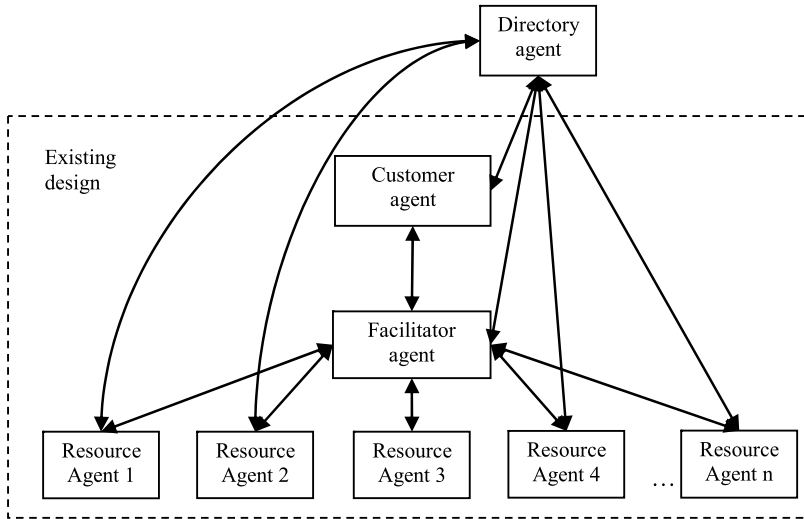


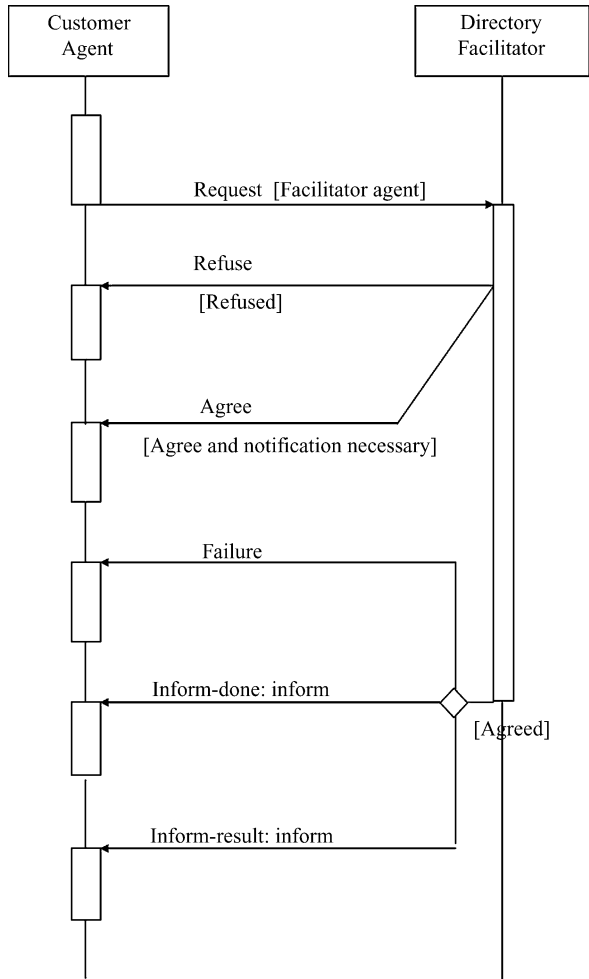
Fig. 3 Information flow in a basic VCIM system (adapted and modified [5])

Table 1 Steps of recourse scheduling for a single order

Step1	When a Facilitator Agent receives a request from a Customer Agent, it divides the order into subtasks.
Step2	The Facilitator Agent sends a request to Directory Agent to find Resource Agents that can perform the subtasks.
Step3	The Directory Agent receives the request and search in its own database. Then return the result to the Facilitator Agent.
Step4	The Facilitator Agent sends a request to Resource Agents that can perform the subtasks.
Step5	The requested Resource Agents generate all possible subtask schedules and return the result to the Facilitator Agent.
Step6	The Facilitator Agent generates all possible production schedules according to received subtask schedules from the Resource Agents.
Step7	The Facilitator Agent finds the best suitable production schedule through a multi-criteria method, and returns the result to the Customer Agent.
Step8	After the Customer Agent receives the order quotations, it selects best suitable Facilitator Agent through a multi-criteria method and confirms the order to that Facilitator.

and sends a request to Directory Agent to find Resource Agents that can perform the subtasks. Then the Facilitator Agent sends requests to Resource Agents returned by Directory. The requested Resource Agents generate all possible subtask schedules and return the results to the Facilitator Agent. The Facilitator Agent generates all possible production schedules according to received subtask schedules from the Resource Agents and finds the best suitable production schedule. After the Customer Agent receives the order quotations, it selects best suitable Facilitator Agent based

Fig. 4 A RIP cycle for Customer Agent searching for Facilitator Agents (adapted and modified from FIPA spec [7])



on their quotation content and other considerable factors. After the Facilitator Agent receives the order confirmation, it sends subtask confirmation to Resource Agents according to the subtask schedule. After the Resource Agent receives the subtask confirmation, it starts the production process.

Multi-agent communication protocol needs to be defined here to support above agent selection and resource scheduling. In this system, when a customer submits a product order, a VCIM agent search and resource scheduling process is initiated. Protocols to support agent communication linkage and information exchange need to be defined. For example, Fig. 4 shows an agent communication protocol used for Customer Agent to search for suitable Facilitator Agents. Here we use Request Interaction Protocol (RIP). The Customer Agent first sends a search request to the Facilitator Agent. The Facilitator Agent replies with either refuse or agree. If agree the Facilitator Agent then process the search based on criteria given by the Customer

Agent. If search fails, it returns Failed. Otherwise it returns Inform-done and Inform-result.

There are other scenarios like Resource Agent registering to Directory agent, Facilitator Agent search in the Directory for suitable Resource Agents, order negotiation between Customer Agent and Facilitator Agent, and order negotiation between Facilitator Agent and Resource Agent. For each scenario, an agent interaction protocol is going to be defined.

3.3 Multi-criteria Selection Integration

The VCIM system purpose is to automatically decompose the customer's order into sub-orders and find best supplier and best paths for parts transport and assembly through computer based resource scheduling. This can be decomposed as two major processes. Process 1 is for Customer Agent to find the best Facilitator Agent. Process 2 is for Facilitator Agent to find the best production schedules by utilizing suitable Resource Agents. Those processes are emulating a customer finding a suitable broker to making some products and the broker finding several suitable workshops to work together for that order.

3.3.1 Factors in VCIM Resource Scheduling

In previous VCIM approach, only delivery time and cost are considered in resource scheduling. While in real world procurement, a lot of other factors affects final decision making. To make VCIM concept more practical, we need think about similar procedures for real person in a company to compare quotation and select outsourcing vendors and factors inside the selection.

According to Kumar [8], there are nine factors to be considered in Vendor Selection. These are price, location, flexible contract terms, cultural match, reputation, existing relationship, commitment to quality, scope of resources, added capability. In order to aim long term supplier relationship, Yao [9] proposed five criteria, cost, quality, project, and certification and delivery performance for the hierarchy. Assessing a group of vendors and selecting one or more of them is a complex task because various criteria must be considered on the decision-making process. Dickson [10] studied the importance of vendor evaluation criteria for industrial purchasing managers and presented 23 vendor attributes that managers consider in such an evaluation, including quality, delivery, price, performance history and others. Weber et al. [11] concluded that quality was the most important factor, followed by delivery performance and price. They found that quality was of 'extreme importance', and delivery was of considerable importance. Hill [13] concluded that quality was an essential factor that qualified a corporation to compete in the marketplace, because vendors with unacceptable quality performance were dropped during the screening phase.

With consideration of previous research on Vendor selection and VCIM, the comparable factors for VCIM resource scheduling are identified as following:

Factors for Process 1, Customer Agent Find and Select the Best Suitable Facilitator Agent for a Particular Order The Customer Agent finds suitable Facilitator Agents in the public registry (aka Directory Facilitator). Then it compares best suitable Facilitator Agent through a multi-criteria method. The comparable factors for this problem are: Credit, Friendship, Price, and Time.

Credit: A property that is stored in the public registry and can be retrieved by request. It is variable and will be adjusted by performance history (rating system)

Friendship: A property that is stored in the Customer Agent. It is variable and will be adjusted by performance history.

Quality: A property that is stored in the public registry and can be retrieved by request. It is variable and will be adjusted by performance history (rating system)

Price: A property that is generated by Facilitator Agent for that particular order. Different Facilitator Agent has different profit margins and different quotation prices from other Resource Agents. Even same production schedule may have different facilitator quotation prices.

Time: A property that is generated by Facilitator Agent for that particular order.

Factors for Process 2, when a Facilitator Agent Receives a Request from a Customer Agent, It Generates all Possible Production Schedules. Then It Finds the Best Suitable Production Schedule Through a Multi-criteria Method The comparable factors for this problem are:

For Cost, Time, Quality, Friendship, and Credit, they are the same as in Process 1. However there is one extra factor used.

Delivery reliability: A property that is stored in the public registry and can be retrieved by request. It is variable and will be adjusted by performance history (rating system).

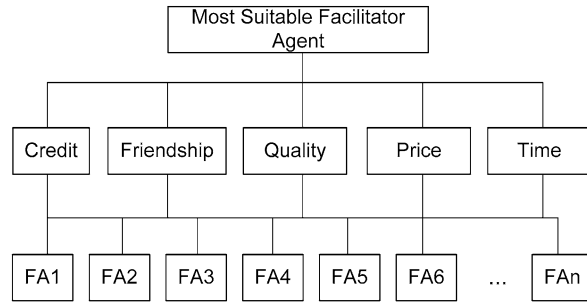
The rating system mentioned above is supposed to be a simple survey of trust and reputation systems like eBay’s feedback form [14]. In this case it is performed by software agents rather than human.

3.3.2 Multi-criteria Selection Process

As multiple factors that have been identified in VCIM resource scheduling, a multi-criteria selection approach is to be integrated. In the recent years, AHP method has been widely adopted as a decision making tool for outsourcing vendor selection problems. Many researches showed that the AHP is very effective solution to different kinds of Multi-criteria vendor selection, such as Manufacturing outsourcing vendor selection [15], Information system outsourcing vendor selection [16], E-business outsourcing vendor selection [17], and 3PL (third-party logistics provider) vendor selection of a 4PL (fourth-party logistics providers) system [18].

As shown in Fig. 5, the hierarchy construction has three layers for Resource Scheduling Process 1. Suitable Facilitator Agents are compared by five factors: Credit, Friendship, Quality, Price and Time to find out the most suitable Facilitator for this process.

Fig. 5 Hierarchy construction model for VCIM Resource Scheduling Process 1



The comparable factors in VCIM are quite similar to those factors in Vendor selection. And some factors we identified are fuzzy, such as quality and friendship. Therefore a Fuzzy AHP method is going to be used in the two VCIM resource scheduling processes mentioned above where multiple factors are involved.

4 Conclusion

VCIM is a way forward to the SMMEs in many countries, where the globalization has impacted dramatically on manufacturing industry. VCIM provides a facility to share resources of partner enterprises that are geographically distributed and provides a competitive edge for SMMEs to be in equal footing with large organizations that have abundance resources to meet the challenges imposed by the globalization. This article discusses an advanced Agent-based VCIM architecture and Multi-criteria Resource Scheduling Process that will help the SMMEs in global competition. As can be seen, future work to do is on optimizing Agent based resource scheduling. This will involve designing new agent behavior, communication protocols, and an effective multi-criteria selection method Model.

References

1. Wang, D., Nagalingam, S.V., Lin, G.C.I.: Development of an agent-based virtual CIM architecture for small and medium manufacturers. *Robot. Comput. Integr. Manufact.* **23**(1), 1–16 (2007)
2. Nagalingam, S.V., Lin, G.C.I.: Latest developments in CIM. *Robot. Comput. Integr. Manufact.* **15**(6), 423 (1999)
3. Wang, D., Nagalingam, S.V., Lin, G.C.I.: Development of a parallel processing multi-agent architecture for a virtual CIM system. *Int. J. Prod. Res.* **42**(17), 3765–3785 (September 2004)
4. Wang, D., Nagalingam, S.V., Lin, G.C.I.: Implementation approaches for a multi-agent virtual CIM system. In: 9th International Conference on Manufacturing Excellence (ICME – 2003), Melbourne, Australia (2003)
5. Wang, D., Nagalingam, S.V., Lin, G.C.I.: A novel multi-agent architecture for virtual CIM system. *Int. J. Agile Manufact. Syst.* **8**(8), 69–82 (2005)

6. Goldman, C.V., Zilberstein, S.: Optimizing information exchange in cooperative multi-agent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia. ACM, New York (2003)
7. Foundation for Intelligent Physical Agents: FIPA request when interaction protocol specification. <http://www.fipa.org/specs/fipa00028/SC00028H.html>
8. Kumar, M., Vrat, P., Shankar, R.: A fuzzy goal programming approach for vendor selection problem in a supply chain. *Comput. Ind. Eng.* **46**(1), 69–85 (2004)
9. Yao, Y., Evers, P.T., Dresner, M.E.: Supply chain integration in vendor-managed inventory. *Decis. Support Syst.* **43**(2), 663–674 (2007)
10. Dickson, G.W.: An analysis of vendor selection systems and decisions. *J. Purchas.* **2**(1), 5–17 (1966)
11. Weber, C.V., et al.: Key practices of the capability maturity model. 1991. CMU/SEI-91-TR-25
12. Zhou, N., Xing, K., Nagalingam, S., Lin, G.: Development of an agent based VCIM resource scheduling process for small and medium enterprises. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 39–44 (2010)
13. Hill, Jr.S.: Some outsourcing successes. *Manufact. Syst.* **18** (6), (2000)
14. Resnick, P., Zeckhauser, R., Swanson, J., Lockwood, K.: The value of reputation on eBay: a controlled experiment. Working Paper for the June 2002 Esa Conference, Boston, Ma, School of Information, University of Michigan. <http://www.si.umich.edu/presnick/papers/postcards/> (2002)
15. Jian-Jun, W., Rui, G., Xin-Jun, D. (eds.): Using a Hybrid Multi-criteria Decision Aid Method for Outsourcing Vendor Selection. IEEE, Piscataway (2008)
16. Fu, Y., Liu, H. (eds.): Information Systems Outsourcing Vendor Selection Based on Analytic Hierarchy Process. Inst. of Elec. and Elec. Eng. Computer Society, Shanghai (2007)
17. Wuwei, L., Yuhong, W., Ao, C. (eds.): Grey Relational Evaluation on Vendor Selection Based on e-Business. IEEE, Piscataway (2008)
18. He, Z., Xiu, L., Wenhua, L., Bing, L., Zhihong, Z. (eds.): An Application of the AHP in 3PL Vendor Selection of a 4PL System. IEEE, Piscataway (2004)

Supreme Court Sentences Retrieval Using Thai Law Ontology

Tanapon Tantisripreecha and Nuanwan
Soonthornphisaj

Abstract This paper presents an improvement of our approach called SCRO_II algorithm. SCRO_I algorithm was initially developed in order to retrieve a set of Supreme Court sentences. The goal of SCRO_I is to provide different law issues among those retrieved documents. We create a new ontology using different semantics to study their performances based on diversity measurement. The contribution of this new ontology is compared to the traditional one. A new procedure is embedded in SCRO_II algorithm to identify a set of synonyms and relations. The experiments were done on Thai Succession Law and Bill of Exchange Law. The experimental results show that SCRO_II outperforms SCRO_I algorithm in both data sets.

Keywords Ontology · Retrieval · Supreme Court sentences · Thai succession law

1 Introduction

Decisions made by judges are very important for users in legal fields because these legal documents provide some legal doctrines that cannot be found in the code laws. Studying Thai law using only code laws is not possible to fulfill the knowledge of those users. The Supreme Court of Thailand had provided a search facility through the Deka System (<http://deka2007.supremecourt.or.th/>). The corpus of the current system consists of 100,010 legal documents. These documents are not predefined into categories of law domains; therefore the retrieval performance is not satisfied in terms of semantic diversity of law content presented in the documents. The reason that legal users need to obtain the variety of law issues from the retrieved document is that they need the complete understanding of each law area in all aspects. To tackle this problem, we propose to use an ontology as a knowledge representation and develop an algorithm called SCRO_I to generate a set of keywords that guarantee

N. Soonthornphisaj (✉)

Department of Computer Science, Faculty of Science Kasetsart University, Bangkok, Thailand
e-mail: fscinws@ku.ac.th

the documents diversity [1]. However, the performance measured in term of document diversity obtained from SCRO_I algorithm is only 54.6%. The objective of this paper is to improve the performance of SCRO_I algorithm. We propose a new ontology model that has an ability to enhance the retrieval performance. Another benefit of our proposed algorithm is that users do not need to input several keywords for searching since the algorithm will automatically create a set of keywords for the retrieval process.

This paper is organized as follows. Section 2 describes an overview of ontology and our proposed ontology. Section 3 addresses the SCRO_I and SCRO_II algorithm. Section 4 describes the evaluation details of ontology-based retrieval system. Finally we conclude our work with some discussion and directions of future work in Sect. 5.

2 Ontology

2.1 *Related Works on Ontology*

Many domain specific ontologies were constructed in various areas such as medical domain [2], e-learning domain [3], legal domain [1, 4]. Ontologies were applied to enhance existing applications such as knowledge-based system [5], computational linguistics [6], etc. Benjamins et al. [7] applied ontology to improve IT support for judges in Spain. They used questionnaires as a tool to collect requirement from users in order to implement the ontology. E-commerce law ontology was constructed using text mining technology by Kayed [8]. A set of conceptual nodes were extracted from e-commerce law cases. They proposed a new algorithm to reduce a number of links connected between nodes in order to decrease the ontology complexity.

Legal Information retrieval research was done for French law [9]. Saravanan et al. did research about query expansion for Indian law in three area which are rent control, income tax and sale tax. They compared their proposed algorithm to the keyword search and found that their algorithm outperformed the traditional retrieval algorithm [10]. The first Korean law ontology was generated using natural language processing technique cooperated with Korean WordNet. The dataset was collected from law text books and research papers about law. After that, their algorithm found a set of word co-occurrences and hand on ability to construct a graph that connects these words together [11].

2.2 *Structure of Ontology*

We set up a structure of the Thai succession law ontology and Thai bill of exchange law ontology using a set of relations and a set of synonyms. Our ontology has two levels, a core level and an extended level ontology.

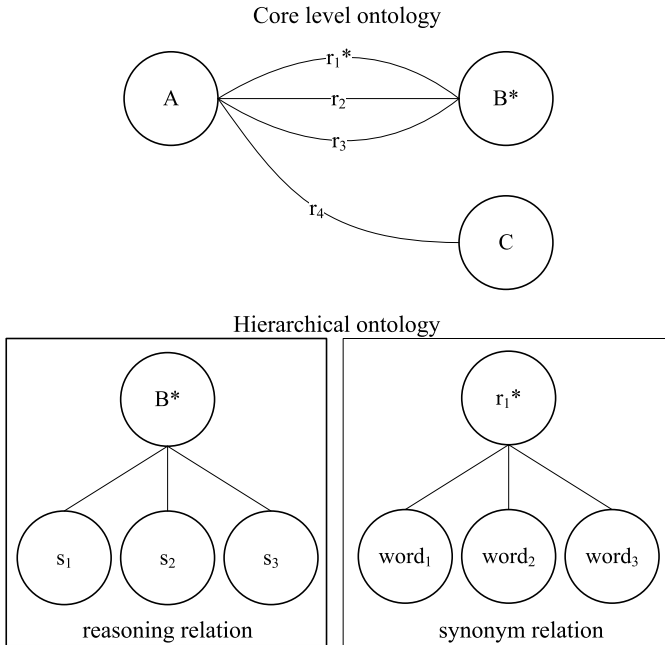


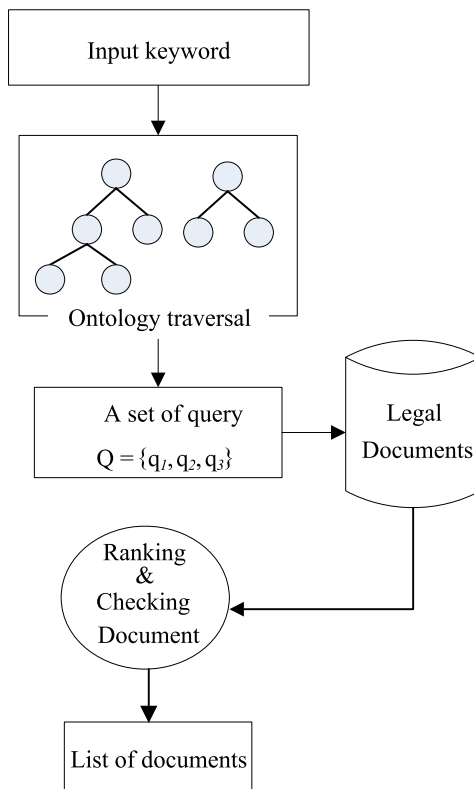
Fig. 1 The proposed framework of ontology is divided into core level and extended level

The core level ontology represents the knowledge of Thai succession law [4]. Figure 1 shows an example of ontology. In case that user enters keyword, ‘ r_1 ’, SCRO_II algorithm traverses through the core level ontology and found the relation ‘ r_1 ’ exists. Next, the algorithm explores all links attached to node between ‘ r_1 ’, (r_1^* , r_2 , r_3 , r_4). The star sign (*) found in r_1 means that there is an extended knowledge related to r_1 . Therefore the algorithm explores the extended level of r_1 and found word₁, word₂ and word₃. Finally a set of queries (Q) are obtained as {A, r_1 , B}, {A, word₁, B}, {A, word₂, B}, {A, word₃, B}, respectively.

In case that the keyword obtained from user is found in the node (e.g. ‘A’) of ontology, SCRO_II traverses through the ontology and generates the set of queries as follows: {A, r_1 , B}, {A, r_2 , B}, {A, r_3 , B}, {A, r_4 , C}, {A, r_1 , S_1 }, {A, r_2 , S_1 }, {A, r_3 , S_1 }, {A, r_1 , S_2 }, {A, r_2 , S_2 }, {A, r_3 , S_2 }, {A, r_1 , S_3 }, {A, r_2 , S_3 }, {A, r_3 , S_3 }, respectively.

We develop an application for a law expert to create Thai succession law ontology and Thai bill of exchange law ontology. First, the conceptual nodes were created when user enter keywords into the system. Then the links were connected when user enter words that represent the relationship between nodes. The synonym and reason relation are automatically created using ThaiLegalWordnet.

Fig. 2 The framework of SCRO_II algorithm



3 Algorithms

3.1 SCRO_II Algorithm

The framework of SCRO_II algorithm is shown in Fig. 2 and Table 1. Given a keyword obtained from user, SCRO_II searches through the core ontology for the keyword. If the word is found in the node of the ontology, the algorithm will traverse through the link to the next node. The words occurred in the nodes and relations are generated to form a set of query.

In case that user enters keyword, 'A', SCRO_II algorithm traverses through the ontology and found the node 'A'. Next, the algorithm explores all links attached with node 'A', ($r_1, r_2, r_3, r_4, \text{word}_1, \text{word}_2, \text{word}_3$). Then a set of queries (Q) are obtained as $\{A, r_1, B\}, \{A, \text{word}_1, B\}, \{A, \text{word}_2, B\}, \{A, \text{word}_3, B\}, \{A, r_2, B\}, \{A, r_3, B\}$ and $\{A, r_4, C\}$, respectively. SCRO_II also explores the extended level of ontology under two different conditions. In case that the keyword is found in the node, the extended level ontology will be explored only when the star sign is found in the node.

On the other hand, if the keyword is the relation found in the ontology, the extended ontology will be explored when the star sign is found on that relation. Then

Table 1 SCRO_II algorithm**Algorithm: SCRO_II**

```

Input: Ontology, keyword
Begin
    Word = {}
    Query = {}
    ConceptType = {NODE, RELATION}
    SearchConcept (Ontology, keyword)
    Word = OntologyTraversal (keyword)
    For Each  $w_i$  in Word Do
        If ConceptType(keyword) = NODE and
            ConceptType( $w_i$ ) = NODE
            If FoundExtendedOntology( $w_i$ )
                ExtendedWord = ExtendedOntologyTraversal()
            If ConceptType( $w_i$ ) = RELATION and
                ConceptType( $w_i$ ) = RELATION
            If FoundExtendedOntology( $w_i$ )
                ExtendedWord = ExtendedOntologyTraversal()
        End For
    Query = QueryGenerate(Word, ExtendWord)
    For each  $q_i$  in Query
         $D_i$  = DocumentRetrieval ( $q_i$ )
        Ranking( $D_i$ )
    End For
    CombineResults = checkDuplicate(D)
End
Output: CombineResults

```

the set of queries are created and used in the retrieval process. Considering a set of retrieved documents, SCRO_II algorithm ranks these documents using *TFIDF* formula (see Eq. 1).

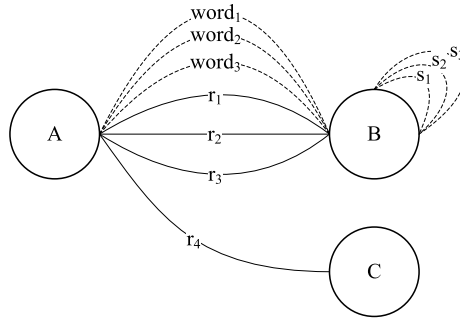
$$w_{in} = tf_{in} \times idf \quad (1)$$

where w_{in} = weight value of word $_i$ in the retrieved document $_n$. tf_{in} = the number of occurrences for word $_i$ in document $_n$.

$$idf = \ln\left(\frac{N}{n_i}\right) \quad (2)$$

where N = the number of retrieved documents. n_i = the number of retrieved documents contain word $_i$.

Fig. 3 The structure of ontology in SCRO_I



SCRO_II algorithm selects the first ranked document as an output for each query and checks for duplicated documents. Since, each query (q_i) is processed independently; therefore the duplicate checking process is done to finalize the output. Because the objective of SCRO_II algorithm is to ensure the diversity of law issues from the final set of retrieved documents.

3.2 SCRO_I Algorithm

The ontology used in SCRO_I algorithm has only one level (Fig. 3; Table 2). Given a keyword, the algorithm explores through the ontology and generates a set of query. The set of query obtained from SCRO_I is different from SCRO_II that yields different performance measure in term of diversity.

In case that the keyword obtained from user is found in the node 'A', SCRO_II traverses through the ontology and generates the set of queries as follows: $\{A, r_1, B\}$, $\{A, r_2, B\}$, $\{A, r_3, B\}$, $\{A, r_4, C\}$, $\{A, word_1, B\}$, $\{A, word_2, B\}$, $\{A, word_3, B\}$, respectively.

If user searches for the keyword, r_1 , the set of queries obtained from SCRO_I are $\{A, r_1, B\}$, $\{A, r_2, B\}$, $\{A, r_3, B\}$, $\{A, word_1, B\}$, $\{A, word_2, B\}$, $\{A, word_3, B\}$, respectively.

3.3 Baseline Algorithm and Deka System

We compare the performance of SCRO_II algorithm with the baseline algorithm and Deka system. The baseline system uses term frequency and inverse document frequency to determine the retrieved document. Whereas the Deka system (Fig. 4) simply uses SQL statements to retrieve legal documents containing a keyword. The current system implemented by the Thai Supreme Court can be accessed via the website www.deka2007.supremecourt.or.th.

Table 2 SCRO_I algorithm

Algorithm: SCRO_I

Input: Ontology, keyword

Begin

Word = {}

Query = {}

Word = OntologyTraversal (keyword)

Query = QueryGenerate (Word, ∅)

For each q_i in Query **Do**

D_i = DocumentRetrieval (q_i)

Ranking (D_i)

End For

CombineResults = checkDuplicate (D)

End

Output: CombineResults

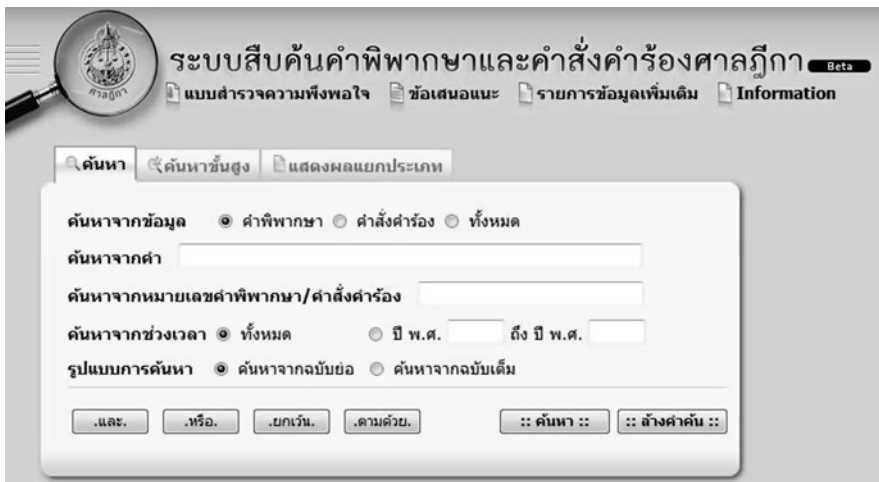


Fig. 4 The supreme court sentences retrieval system (Deka)

4 Experiments

4.1 Data Set

We collect all judge sentences related to Thai succession law and Thai bill of exchange law. There are 248 and 200 documents, respectively, in the corpus. The word segmentation is performed since Thai language has no explicit word boundary.

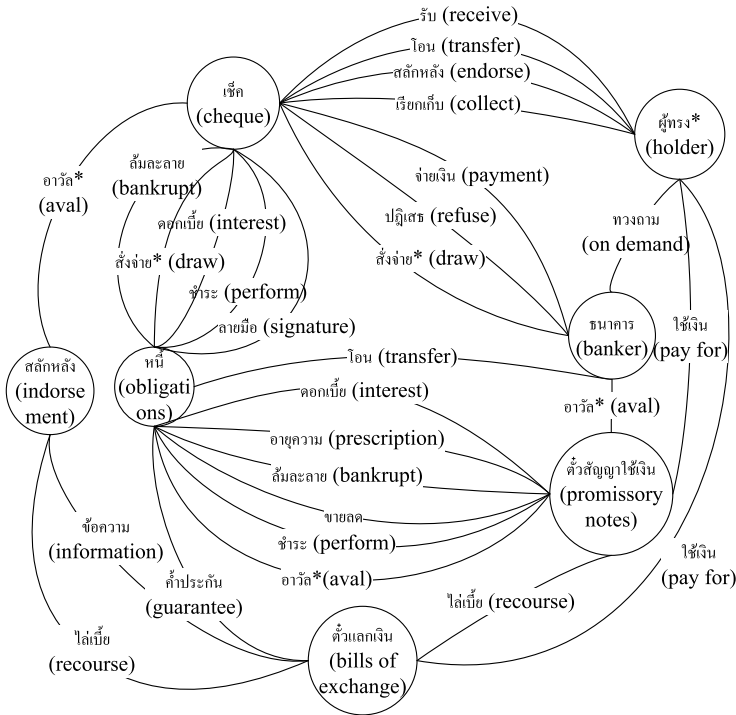


Fig. 6 Thai bill of exchange law ontology

In order to evaluate the retrieval performance on Thai succession law, we set up the experiment using 12 keywords which are มรดก (estate), ผู้จัดการมรดก (administrator of an estate), ทายาท (heir), ผู้ตาย (deceased), พยาน (witness), พินัยกรรม (wills), เจ้าหนี้ (creditor), กำจัดมิให้รับมรดก (exclusion for the succession), สละมรดก (renunciation of an estate), พินัยกรรมไม่สมบูรณ์ (incomplete will), ชำระหนี้ (performing loan), ตั้งผู้จัดการมรดก (appointment of the administrator of an estate), respectively.

In case of Thai bill of exchange Law, we set up the experiment using 12 keywords which are เช็ค (cheque), ตั๋วแลกเงิน (bills of exchange), ตั๋วสัญญาใช้เงิน (promissory notes), ผู้ทรง (holder), ธนาคาร (banker), หนี้ (obligations), สลักหลัง (endorse), ดอกเบี้ย (interest), โอน (transfer), ไล่เบี้ย (recourse), อวัล (aval), ส่งจ่าย (draw), respectively.

4.3 Experimental Results

Considering Tables 3 and 4, we found that SCRO_II can retrieve a set of legal documents with more diversity than those of SCRO_I. In Table 3, the most difficult

Table 3 The comparison of performance in term of the number of law issues found by algorithms using sets of query on Thai succession law data set

Keyword	Number of law issues	Number of law issues found by algorithms			
		SCRO_II	SCRO_I	Baseline	Deka
มรดก (estate)	19	14	8	8	10
ผู้จัดการมรดก (administrator of an estate)	19	10	10	6	6
ทายาท (heir)	17	14	9	9	8
ผู้ตาย (dead person)	4	4	3	1	2
พยาน (witness)	3	3	3	2	1
พินัยกรรม (wills)	9	6	9	2	2
เจ้าหนี้ (creditor)	7	4	3	1	2
ก้ำจัดมิให้รับมรดก (exclusion for the succession)	6	4	2	2	6
สละมรดก (renunciation of an estate)	3	2	2	1	3
พินัยกรรมไม่สมบูรณ์ (incomplete will)	4	3	2	2	1
ชำระหนี้ (performing loan)	3	2	1	1	1
ตั้งผู้จัดการมรดก (appointment of the administrator of an estate)	5	3	3	2	1

keyword for algorithms are “estate” and “heir” because they serve as a general term of the domain, so retrieving a set of diversified documents for such keyword is considered to be hard. For the keyword, “estate”, SCRO_II is able to retrieve a set of documents that have 14 issues whereas SCRO_I gets only 8 issues. For the keyword, “heir”, SCRO_II gets 14 different issues from a set of retrieved documents whereas SCRO_I gets 9 issues.

In Table 4, the most difficult keyword for algorithms is “holder” because they serve as a general term of the domain, so retrieving a set of diversified documents for such keyword is considered to be hard. For the keyword, “holder”, SCRO_II is able to retrieve a set of documents that have 8 issues whereas SCRO_I gets only 4 issues.

SCRO_II and SCRO_I can completely retrieve all issues related to the keyword, ผู้ตาย (dead person) and พยาน (witness). Moreover, the performance of SCRO_II and SCRO_I on keyword, ผู้จัดการมรดก (administrator of an estate) and สละมรดก (renunciation of an estate) are the same.

Table 5 provides details about the experiments to see the performance of algorithms using different types of query, i.e. Query_A and Query_B. Query_A contain

Table 4 The comparison of performance in term of the number of law issues found by algorithms using sets of query on Thai bill of exchange data set

Keyword	Number of law issues	Number of law issues found by algorithms			
		SCRO_II	SCRO_I	Baseline	Deka
เช็ค (cheque)	13	9	10	2	7
ตั๋วแลกเงิน (bills of exchange)	5	4	3	1	3
ตั๋วสัญญาใช้เงิน (promissory notes)	9	6	7	3	7
ผู้ทรง (holder)	12	8	4	3	4
ธนาคาร (banker)	5	4	5	2	4
หนี้ (obligations)	13	9	10	6	5
สลักหลัง (endorse)	3	3	2	2	2
ดอกเบี้ย (interest)	2	2	2	1	1
โอน (transfer)	3	1	1	1	1
ไล่เบี้ย (recourse)	2	2	1	1	1
อาวัล (aval)	6	5	4	1	3
สั่งจ่าย (draw)	2	2	1	1	1

Table 5 Comparison between types of query in each algorithm

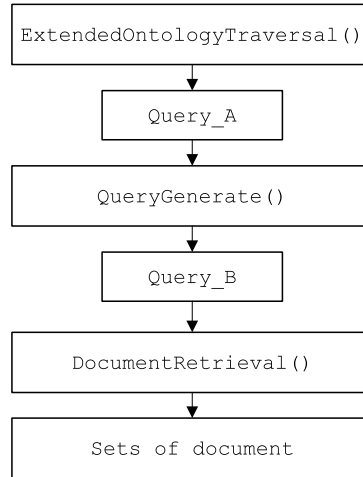
Algorithm	Query_A		Query_B	
	Succession Law	Bill of exchange Law	Succession Law	Bill of exchange Law
SCRO_II	0.34	0.33	0.72	0.79
SCRO_I	0.35	0.37	0.59	0.66
Baseline	0.30	0.32	0.37	0.37
Deka	0.34	0.25	0.45	0.26

all keywords getting from `ExtendedOntologyTraversal()` in `SCRO_II` algorithm. On the other hand, `Query_B` are set of query obtain from `QueryGenerate()`.

The performance measured on diversity of documents shows that the average performance of `SCRO_II` is higher than that of `SCRO_I`. `SCRO_II` gets 72%, whereas `SCRO_I` gets 59%. Because the two-level ontology is able to expand sets of query more than the ontology of `SCRO_I` which has only one level.

Consider Thai succession law data set, the performance measured on diversity of documents using `Query_A`, `SCRO_II` and `SCRO_I` get 34% and 35% respectively, whereas the performance of `SCRO_II` and `SCRO_I` using `Query_B` get 72% and 59% respectively. For Thai bill of exchange Law data set, the performance measured on diversity of documents of `Query_A`, `SCRO_II` and `SCRO_I` get 33% and

Fig. 7 Query expansion process in SCRO_II



37% respectively, whereas the performance measured on diversity of documents of Query_B, SCRO_II and SCRO_I get 79% and 66%, respectively.

Hence, using Query_B to retrieve documents can obtain diversify documents for each issue but Query_A has less performance in retrieving such diversify documents because a single set does not represent each issues.

Consider ontology shown in Fig. 1, if user enter keyword 'A' then Extended-OntologyTraversal() returns a query (name as Query_A) $\{r_1, r_2, r_3, r_4, \text{word}_1, \text{word}_2, \text{word}_3\}$. On the other hand, QueryGenerate() returns sets of query (Query_B) $\{A, r_1, B\}, \{A, \text{word}_1, B\}, \{A, \text{word}_2, B\}, \{A, \text{word}_3, B\}, \{A, r_2, B\}, \{A, r_3, B\}$ and $\{A, r_4, C\}$, respectively.

Consider Fig. 7, QueryGenerate() plays an important role in the document retrieval process. Using Query_A in Thai Succession law data set, SCRO_II, SCRO_I, Baseline and Deka system get 34%, 35%, 30% and 34% respectively. Whereas SCRO_II, SCRO_I, Baseline and Deka system using Query_B get 72%, 66%, 37% and 26% respectively. Because each subset of Query_B is considered a law issue, but Query_A cannot exhibits the law issue diversity.

Considering the baseline algorithm, we surprisingly found that the Deka system using SQL statement outperforms the traditional information retrieval system using *TFIDF* formula. The reason is that Deka system simply retrieves judges sentences having the keyword and sorts by the document ID no. It is assumed that the succession law cases submitted to the Supreme Court are likely to have different issues. However, the Deka system has less diversity value than SCRO_II because the criteria to determine the performance of any information retrieval system concerns about the ranking process. We want the system to deliver the suitable number of choices (documents) to users so that they can effectively view their search results. Therefore, the objective of our research is to develop an algorithm to effectively retrieve a suitable number of judge sentences that guarantee the variety of law issues.

5 Conclusions

This study proposes an ontological approach for Supreme Court Sentence Retrieval on Thai succession law data set and Thai bill of exchange law dataset. The objective of using ontology is to provide a set of diversify documents for user. We found that using hierarchical style of ontology can provide more variety of law issues in a set of retrieval results. The experimental results demonstrate that SCRO_II outperforms SCRO_I algorithm in term of diversity value because ontology structure can expand sets of query which have a diversity of law issues.

Comparing between type of query, i.e. Query_A and Query_B, we found that sets of query exhibit diversity results more than a set of query. Beside, users are satisfied with the judge documents retrieved by SCRO_II.

Acknowledgements This research was partly supported by Faculty of Science and The Graduate School of Kasetsart University.

References

1. Tantisiripreecha, T., Soonthornphisaj, N.: Query expansion algorithm for supreme court sentences retrieval using ontology. In: Proceedings of the 48th Kasetsart Annual Conference, pp. 43–50 (2009)
2. Jovic, A., Prcela, M., Gamberger, D.: Ontologies in medical knowledge representation. In: Proceedings of the 29th International Conference on Information Technology Interfaces, pp. 535–540 (2007)
3. Henze, N., Dolog, P., Nejdil, W.: Reasoning and ontologies for personalized e-learning in the semantic web. *Educ. Technol. Soc.* **7**(4), 82–97 (2004)
4. Tantisiripreecha, T., Soonthornphisaj, N.: A study of Thai succession law ontology on supreme court sentences retrieval. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 146–151 (2010)
5. Valente, A., Breuker, J.: Making ends meet: conceptual models and ontologies in legal problem solving. In: Proceedings of the XI Brazilian AI Symposium, pp. 1–15 (1994)
6. Hohfeld, W.: *Fundamental Legal Conceptions as Applied in Legal Reasoning*. Yale University Press, London (2008)
7. Benjamins, V.R., Contreras, J., Casanovas, P., Ayuso, M., Becue, M., Lemus, L., Urios, C.: Ontologies of professional legal knowledge as the basic for intelligent IT support for judges. In: *Artificial Intelligence and Law*, pp. 359–378 (2006)
8. Kayed, A.: Building e-laws ontology: new approach. In: *OTM Workshops*, pp. 826–835 (2005)
9. Guiraude, L., Sylvie, D.: Updating ontology in the legal domain. In: *Proceedings of the 10th Int. Conf. on Artificial Intelligence and Law* (2005)
10. Saravanan, M., Ravindran, B., Raman, S.: *Improving Legal Information Retrieval Using an Ontological Framework*. Springer Science Business Media B.V., Berlin (2009)
11. Soonhee, H., Youngim, J., Aesun, Y., Kwon, H.C.: Building Korean classifier ontology based on Korean WordNet, pp. 261–268 (2006)

Genetic Algorithm Based Model for Effective Document Retrieval

Hazra Imran and Aditi Sharan

Abstract One central problem of information retrieval is to determine the relevance of documents with respect to the user information needs. The choice of similarity measure is crucial for improving search effectiveness of a retrieval system. Different similarity measures have been suggested to match the query and documents. Some of the popular measures being: cosine, jaccard, dice, okapi etc., each having their own pros and cons. Accordingly one may give better result over other depending on users need, document corpus, organization and indexing of corpus. Therefore it may be justifiable to combine these measures and develop a new similarity measure which can be named as combined similarity measure. Now individual measures can be assigned weights in different proportion in combined similarity measure. In order to optimize ranking of relevant documents, individual weights have to be optimized. In this chapter we suggest a genetic algorithm based model for learning weights of individual components of combined similarity measure. We have considered two different types of functions viz: non-order based and order based fitness functions to evaluate the goodness of the solution. A non-order based fitness function is based on recall-precision values only. However, it has been observed that a better fitness function can be obtained if we also consider the order in which relevant documents are retrieved. This leads to an idea of order based fitness functions. We evaluated the efficacy of a genetic algorithm with various fitness functions. The experiments have been carried out on TREC data collection. The results have been compared with various well-known similarity measures.

Keywords Document ranking · Genetic algorithms · Similarity measures · Information retrieval · Vector space model

H. Imran (✉)

Department of Computer Science, Jamia Hamdard, Hamdard Nagar, New Delhi 110 062, India
e-mail: himran@jamiahamdard.ac.in

1 Introduction

An Information Retrieval (IR) is one that accepts a user query and returns a ranked list of relevant documents based on users query. The efficiency of IR system is entirely dependent on similarity measure used for ranking the documents. Thus choice of appropriate similarity measure is most crucial aspect for developing an efficient IR system. Different similarity measures have been suggested for matching query and document, each having their own pros and cons. Sometimes it may be justifiable to combine various similarity measures leading to a new similarity measure, which can be called as combined similarity measure (CSM). Individual measures in CSM can be given weight in different proportions. In order to obtain optimal ranking weights of these individual measures need to be optimized. In this chapter we suggest a GA based model for efficient matching of query and documents using combined similarity measure. The chapter is divided in 6 sections. In Sect. 2 we briefly introduce IR related terminology, which includes Vector Space Model (VSM), similarity measures, and measures for evaluating IR system. In Sect. 3 we introduce genetic algorithm and its functioning. Section 4 deals with our proposed model. In Sect. 5 we present experiments and results. Finally Sect. 6 concludes our work.

2 Information Retrieval System

Information retrieval system is devoted to finding “relevant” documents with respect to users query. Standard IR systems are based on Boolean [1], vector [6], and probabilistic models [8]. Each model describes documents, queries and provides algorithms to compute similarity between user’s query and documents. The objective of an information retrieval system is to provide its users with satisfactory retrieval results.

2.1 Vector Space Model

Our proposed model is based on Vector Space Model (VSM). In VSM, both documents and the user given query are represented as vector of terms. Suppose there are t index terms in a collection of documents. Then document D_i and query Q can be represented as

$$D_i = (d_{i1}, d_{i2}, \dots, d_{it}) \quad (1)$$

$$Q = (w_{q1}, w_{q2}, \dots, w_{qt}) \quad (2)$$

where d_{ij} ($j = 1$ to t) are term weights in document D_i and w_{qj} ($j = 1$ to t) are term weights in the query Q . There are different methods of assigning weights to d_{ij} . Representative weighting functions include term frequency (TF), inverse document frequency (IDF), the product of TF and IDF (TF·IDF).

2.1.1 Term Frequency

The TF method assumes that high frequency content bearing terms represent the main topics of the document, and measures the frequency of occurrence of an index term in the document text as its weight, that is,

$$TF_{ij} = \text{Frequency of } j\text{th term in document } i \quad (3)$$

2.1.2 Inverse Document Frequency

However, terms may occur frequently in a document yet be poor indicators of its content. For example, the term computer is not a good index term for a document collection in computing. The fewer the texts in which a term appears, the more discriminating that term is. Therefore, the weight of a term should be inversely related to the number of documents in which the term appears. The IDF defined by Spark Jones [10] is commonly used to incorporate this effect; it is computed as

$$idf_j = \frac{\log N}{n_j} \quad (4)$$

where N is the number of documents in the reference collection and n_j is the number of documents in the reference collection having index term j . Within a document collection, the best terms for a particular document will be those occurring frequently in that document but rarely in the other documents.

2.1.3 Term Frequency-Inverse Document Frequency Weight

The above findings form the basis for a very popular term weighting function that determines the product of the TF and the IDF (TF · IDF) [7, 9] of the index term:

$$d_{ij} = TF_{ij} \times idf_j \quad (5)$$

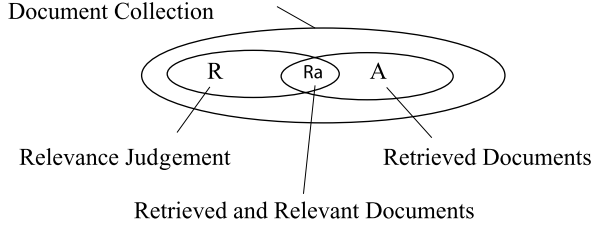
It captures the essence of two main assumptions:

1. A word with high term frequency (i.e. that occurs many times in the document) is more likely to describe the contents of the document than a word that only occurs one or a few times, and
2. A word with high document frequency (i.e. that occurs in many documents) is less likely to discriminate between relevant and non-relevant documents than a word that only occurs in a few documents.

2.2 Ranking of Documents

Once documents and query are arranged in VSM, a vector based similarity measure can be used for matching query and document. For calculating similarity of query

Fig. 1 Precision and Recall for a given example information request



and document cosine, jaccard and dice are the popularly used similarity measures. Cosine, jaccard and dice similarity measures are defined as follows:

$$\text{Cos}(Q, D_i) = \frac{\sum_{j=1}^t w_{qj} d_{ij}}{\sqrt{\sum_{j=1}^t (d_{ij})^2 \sum_{j=1}^t (w_{qj})^2}} \quad (6)$$

As the angle between the vectors shortens, the cosine angle approaches 1, meaning that the two vectors are getting closer, meaning that the vectors represent the similarity of document and query increases. The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the document and query vectors.

$$\text{Jaccard}(Q, D_i) = \frac{\sum_{j=1}^t w_{qj} d_{ij}}{\sum_{j=1}^t (d_{ij})^2 + \sum_{j=1}^t (w_{qj})^2 - \sum_{j=1}^t w_{qj} d_{ij}} \quad (7)$$

For document and query vector, the dice coefficient may be defined as twice the shared information (intersection) over the combined set (union)

$$\text{Dice}(Q, D_i) = \frac{2 \sum_{j=1}^t w_{qj} d_{ij}}{\sqrt{\sum_{j=1}^t (d_{ij})^2 \sum_{j=1}^t (w_{qj})^2}} \quad (8)$$

2.3 Evaluation

Recall and precision are basic evaluation parameters (Fig. 1). For an example information request I , a set of relevant documents R is given, with $|R|$ being the number of documents in this set. The evaluated retrieval strategy yields the document set A for the example request, likewise $|A|$ being the number of documents in the set. Let $|R_a|$ be the number of documents in the intersection of these two sets.

Definition Recall indicates the relevant documents, which have been retrieved.

$$\text{Recall} = \frac{|R_a|}{|R|} \quad (9)$$

Definition Precision indicates how many retrieved documents are relevant.

$$\text{Precision} = \frac{|R_a|}{|A|} \quad (10)$$

Recall and precision evaluate IR system along different dimensions. Precision measures the accuracy of the result in topmost ranked documents whereas recall measures quality of overall answer and breadth of algorithm. In many situations, the use of single measure which combines recall and precision could be more appropriate. One such measure is F measure

$$F\text{-measure} = \frac{2 * \text{Recall} * \text{Precision}}{(\text{Recall} + \text{Precision})} \quad (11)$$

This is also known as the F_1 measure, because recall and precision are evenly weighted.

3 Genetic Algorithms

The genetic algorithms represent an artificial intelligence search technique that emulates the process of the natural selection. As input they have a population of individuals known as *chromosomes*, which represent the possible solutions to the problem. Initially these inputs are randomly generated, although if there is some knowledge available concerning the said problem, it can be used to create part of the initial set of potential solutions [4]. These individuals evolve in successive iterations known as *generations*, by means of processes of *selection, crossover, and mutation*. These iterations halt when the system no longer improves, or when a preset maximum number of generations is reached. The output of the GA will be the best individual of the end population, or a combination of the best chromosomes of that population. GA requires the evaluations of the fitness function to assign a utility value to every solution produced.

Genetic Algorithms are learning algorithms as well as offer a domain independent search ability that can be used in many learning tasks. Due to this reason, the application of GAs to IR has increased in the last decade [3, 5].

4 Our Approach

In our problem, we have defined a combined similarity measure [2] based on genetic algorithm consisting of standard similarity measures.

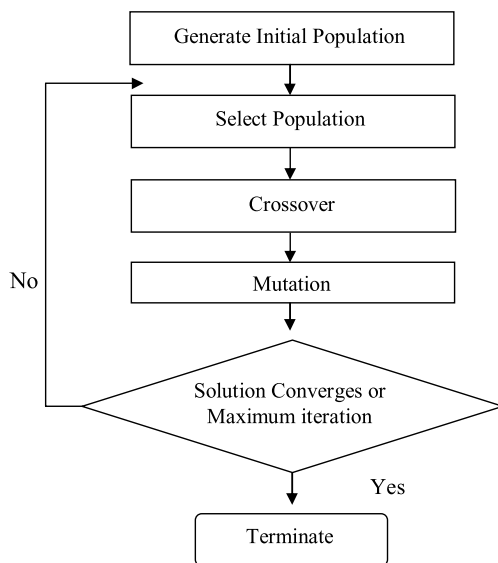
4.1 Combined Similarity Measure Using Genetic Algorithm

Our combined similarity measure is a weighted sum of the scores returned by different matching measures.

In general a combined similarity measure is represented as:

$$\text{combined_similarity_measure}(D, Q) = \sum_{i=1}^M (wt_i \times SM_i(D, Q)) \quad (12)$$

Fig. 2 Genetic algorithm process



where $SM_i(D, Q)$ signifies the score of document D for given query Q for i th similarity measure. M is total number of standard similarity measures considered. We have used three similarity measures: $\cos(SM_1)$ and $\text{jaccard}(SM_2)$ and $\text{dice}(SM_3)$. wt_i is the weight assigned to i th similarity measure. Weights wt_1 and wt_2 range from 0.0 to 1.0. A less weight signifies that associated similarity measure is less significant. We assign appropriate weights to the components of these measures so as to achieve maximum retrieval efficiency.

In next subsection we present our genetic algorithm based model for finding optimal weights using combined similarity measure (Fig. 2).

4.2 Proposed Model

Figure 4 presents our proposed model. The input to the model is document and query and output is the optimized weights for combined similarity measure. The working of module is as below:

1. Documents and queries are preprocessed and tokenized. During preprocessing we perform stop word removal and stemming. The result obtained is a collection of tokens.
2. Documents and queries are indexed using Vector Space Model. We assigned weights to the terms in each document by the classical *tf.idf* scheme.
3. Documents are matched with queries with different similarity measures. We have used similarity measures: \cos , jaccard and dice . These values are returned as SM_1, SM_2, \dots, SM_n

Fig. 3 Chromosome representation

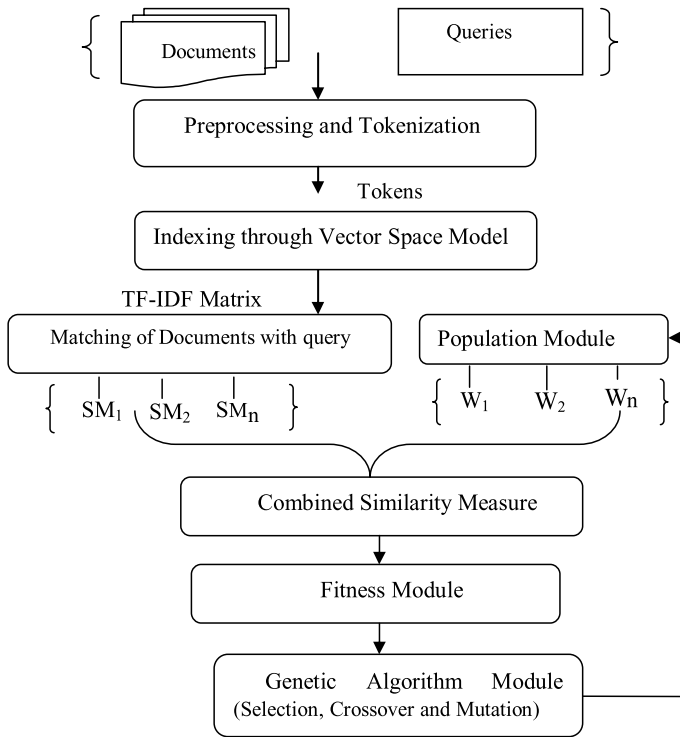
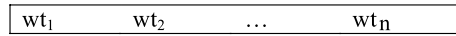


Fig. 4 Our proposed genetic algorithm based model

4. The population module generates chromosomes for initial population of genetic algorithm. The chromosome is represented in the following way (see Fig. 3), where w_{t_i} = weight of the i th similarity measure. We have used a real-valued chromosome because it is more natural representation for our problem and it decreases dramatically the number of genes required to specify a design, thus making the solution space easier to search.
5. Considering similarity measures and weights as input, combined similarity module calculates similarity of the document with respect to combined similarity measure using Eq. 12.
6. Fitness module is used to find the fitness of the solution. Fitness function is a performance measure that is used to evaluate how good each solution is. Given a chromosome, the fitness function must return a numerical value that represents the chromosome's utility. Previous work has been done in GA considering fitness functions based on recall and precision only. However it has been observed that a better fitness function can be obtained if we also consider the order of the retrieval of documents. Order-Based Fitness function takes into account the

number of relevant and irrelevant documents along with the order of their appearance. Note that the documents retrieved earlier in order have higher utility in comparison to documents retrieved later. The basic idea being that it is not the same that the relevant documents appear at the beginning or at the end of the list of retrieved documents. We have used following non-order and order-based fitness functions.

$$\text{Fitness1}_{\text{non-order-based}} = \frac{(2 \times \text{recall} \times \text{precision})}{\text{recall} + \text{precision}} \quad (13)$$

$$\text{Fitness2}_{\text{order-based}} = \frac{1}{|D|} \sum_{i=1}^{|D|} \left(r(d_i) \sum_{j=i}^{|D|} \frac{1}{j} \right) \quad (14)$$

where $|D|$ is the total number of documents retrieved, and $r(d)$ is the function that returns the relevance of document d , giving 1 if the document is relevant and a 0 otherwise. The importance of the order-based fitness function can be justified with the following example. Let us assume that two similarity measures retrieve 6 relevant documents in the top 15 result giving Case 1 and Case 2, where

Case 1 = 100100100111000

Case 2 = 111010110000000

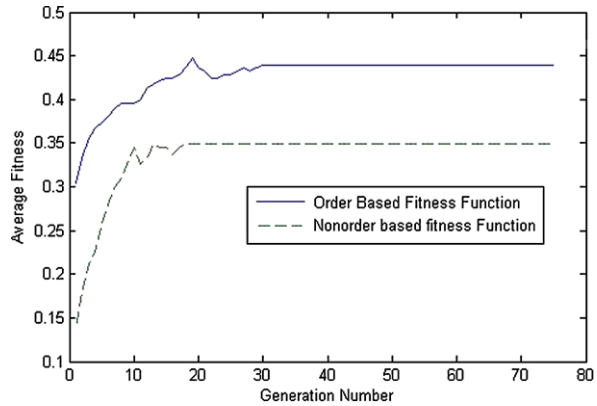
The relevance information is coded as 1 (if relevant) and 0 (if irrelevant). Here we observe that the recall and precision in both the cases is same. However one can easily see that Case 2 has a better performance as all the 6 documents are retrieved earlier in comparison to Case 1. If we consider the total number of relevant document as 10 then the values for $\text{Fitness1}_{\text{non-order-based}}$ is .48 for both the cases whereas $\text{Fitness2}_{\text{order-based}}$ are calculated as 0.45 for Case1 and 0.64 for Case2. Therefore we can justify that $\text{Fitness2}_{\text{order-based}}$ is better.

The fitness module sorts the population on the basis of the combined similarity measure obtained. Further precision and recall are calculated and finally the fitness is calculated for each member of the population using Eqs. 7 and 8. Once the fitness is calculated next modules applies standard GA functions: selection, crossover and mutation and generates new population. The whole process is repeated iteratively until population converges or maximum number of iterations has been carried out.

7. Genetic Algorithm module applies the standard GA functions (selection, crossover, mutation) to generate new population from the old population, which is discussed as follows.

- Selection: – Selection embodies the principle of ‘survival of the fittest’. Chromosomes having higher fitness are selected for crossover. The roulette wheel reproduction process [11] was used to select individuals.
- Crossover: – Crossover is the genetic operator that combines two chromosomes together to form new chromosome. We have used single point crossover where a locus position is selected within two parent chromosomes and the genes are swapped from that position to the end of parent.

Fig. 5 Curve showing variation of average fitness with generation number for both the fitness function



- **Mutation:** – Mutation involves the modification of the values of each gene of a solution with some probability (mutation probability).

5 Experiments and Results

For our experiments, we used volume 1 of the *TIPSTER* document collection, a standard test collection in the IR community. Volume 1 is a 1.2 Gbyte collection of full-text articles and abstracts, divided into seven main files. The documents came from the following sources.

1. WSJ – Wall Street Journal (1986, 1987, 1988, 1989, 1990, 1991 and 1992)
2. AP – AP Newswire (1988, 1989 and 1990)
3. ZIFF – Information from Computer Select disks (Ziff-Davis Publishing)
4. FR – Federal Register (1988)
5. DOE – Short abstracts from Department of Energy

The experiments were done for non-order-based (Fitness1nonorderbased) and order based (Fitness2orderbased) fitness functions. The control parameters set empirically were as follows: population size = 100, probability of crossover ($P_c = 0.7$), and probability of mutation ($P_m = 0.01$). The document cut off was 10. We have performed our experiment on 50 queries. Below is the example of applying Genetic Algorithm based model on TREC query 51 [*Airbus Subsidies*].

Individual [0.862747, 0.064331, 0.656781] Fitness: 0.602898
 Individual [0.862747, 0.064331, .6567812] Fitness: 0.605797
 Individual [0.862747, 0.064331, 0.656781] Fitness: 0.608695
 Individual [0.702370, 0.237541, 0.921758] Fitness: 0.602898

Below are the graphs to show the results of experiment.

Fig. 6 Curve showing recall and precision graph of cos, Jaccard, Dice and combined similarity measure

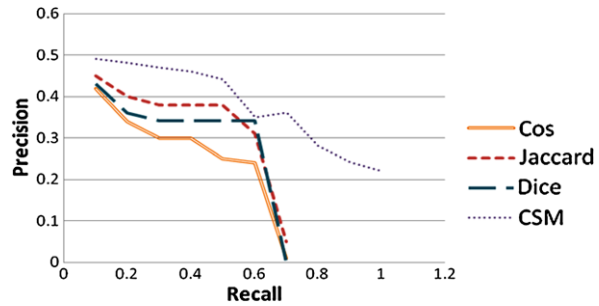


Figure 5 shows variation of average fitness with generation number for a specific query 51 of TREC dataset. As shown the average fitness increases with generation number. On the basis of our experiment we observed that for 55% and 60% of the queries of TREC dataset has increased average fitness in successive generations. We compared the results of our experiments with the recall and precision values obtained by all three standard similarity measure i.e. Cos, Jaccard Dice and combined similarity measure. For almost all the queries our experiment gave better results. Figure 6 shows recall precision curve for all the three standard measures and CSM for a specific query in TREC. Document cutoff was 10. Recall-precision curve is a standard curve for measuring efficiency of any retrieval algorithm. They are useful because they allow us to evaluate quantitatively both quality answer and breadth of retrieval algorithm. Good quality information retrieval algorithm gives high precision at low recall values. One can easily observe in Fig. 6 that initially all similarity measures start with reasonable precision values (between 0.4 to 0.5). However standard similarity measures show a sharp decline in the beginning whereas our combined similarity measure shows gradual decrease in precision. This indicates that combined similarity measure gives better quality of result. The breadth of combined similarity measure is maximum as it is giving a recall value 1.

6 Conclusion

In this paper we have developed a GA based model for efficient matching of query and documents by an Information Retrieval System. An Information Retrieval system may use different similarity measures for matching query and document. Each of these measures have their own pros and cons. Choice of appropriate similarity measure is most crucial aspect that determines efficiency of an IR system. In this work we suggest use of combined similarity measure, which combines individual similarity measures giving them appropriate weights in order to obtain optimally ranked document set. GA has been used for learning these optimum weights. We have used two different type of fitness functions viz: non-order and order based fitness function. Extensive experiments have been performed on TREC data set. As expected GA improves efficiency of IRS, order based fitness being more efficient. This work provides a framework for exploring use of Genetic algorithms to improve efficiency of an Information Retrieval System.

References

1. Bookstein, A.: Probability and fuzzy-set applications to information retrieval. *Ann. Rev. Inform. Sci. Technol.* **20**, 117–151 (1985)
2. Imran, H., Sharan, A.: A framework for efficient document ranking using order and non-order based fitness function. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 71–76 (2010)
3. Lourdes, A., Jose, R.: Improving query expansion with stemming terms: a new genetic algorithm approach. In: *Evolutionary Computation in Combinatorial Optimization* (2008)
4. Michalewicz, Z.: *Genetic Algorithms+Data Structures = Evolution Programs*. Springer, Berlin (1996)
5. Pérez-Agüera, J.R.: Using genetic algorithms for query reformulation. In: *BCS IRSG Symposium: Future Directions in Information Access (FDIA 2007)* (2007)
6. Robertson, S.E.: The probabilistic character of relevance. *Inf. Process. Manag.* **13**, 247–251 (1997)
7. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. In: Jones, K.S., Willett, P. (eds.) *Readings in Information Retrieval*, pp. 323–328. Morgan Kaufman Publishers, San Francisco (1997)
8. Salton, G., McGill, M.: *Introduction to Modern Information Retrieval*. McGraw-Hill, New York (1983)
9. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. *Commun. ACM* **26**(12), 1022–1036 (1983)
10. Spark Jones, K.: A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **28**(1), 11–22 (1972)
11. Vrajitoru, D.: Crossover improvement for genetic algorithms in information retrieval. *Inf. Process. Manag.* **34**(4), 405–415 (1998)

An Agent-Based Cloud Service Discovery System that Consults a Cloud Ontology

Taekyeong Han and Kwang Mong Sim

Abstract This paper presents a Cloud service discovery system (CSDS) that aims to support Cloud users in finding a Cloud service over the Internet. The CSDS interacts with a Cloud ontology to determine the similarities among services. The significance of this project is that it is the first attempt in building an agent-based discovery system that consults an ontology when retrieving information about Cloud services. One of the main contributions of this work is building a Cloud service reasoning agent (CSRA) that enables the CSDS to (1) reason about the relations of Cloud services and (2) rate the search results. Another contribution of this work is designing and constructing a Cloud ontology consisting of a taxonomy of concepts of Cloud services that enables the CSRA to determine the relations of Cloud services using three service reasoning methods: (1) similarity reasoning, (2) equivalent reasoning, and (3) numerical reasoning. Empirical results show that using the Cloud ontology, the CSDS is more successful in finding Cloud services that are closer to users' requirements. Experiments are also conducted to examine the effect of using different combinations of the three service reasoning method: (1) using only similarity reasoning, (2) using similarity reasoning and equivalent reasoning, and (3) using all three reasoning methods. Additionally, a proof-of-concept example demonstrates the major functionalities of the CSDS.

Keywords Cloud computing · Cloud ontology · Software agent · Web information retrieval

K.M. Sim (✉)

School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, South Korea
e-mail: kmsim@gist.ac.kr

1 Introduction

Cloud computing is Internet (Cloud) based development and the use of computer technology (computing) whereby dynamically scalable and often virtualized resources are provided as a service over the Internet [1]. Consumers of Cloud computing will not compute using their own computer, but move their programs and data to the Clouds consisting of computation and storage utilities provided by third parties. Cloud computing providers publish Cloud services over the Internet, and consumers normally access these services provided by Cloud application layer through web-portals [2].

To date, however, there is no discovery mechanism for searching different kinds of Clouds. Cloud consumers generally have to search for appropriate Cloud services manually [3]. Even though there are many existing generic search engines that consumers can use for finding Cloud services, these engines may return URLs containing not relevant web-pages to meet the original service requirements of consumers. Intuitively, visiting all the web-pages can be a time-consuming job. Whereas generic search engines (e.g., Google, MSN, etc.) are very effective tools for searching URLs for generic user queries, they are not designed to reason about the relations among the different types of Cloud services and determining which service(s) would be the best or most appropriate service for meeting consumers' service requirements. Hence, service discovery mechanisms for reasoning about similarity relations among Cloud services are needed. The significance of this work is that to the best of the authors' knowledge, it is the earliest effort in constructing a Cloud service discovery system (CSDS) to assist users in searching for Cloud services more efficiently. However, it is noted here that this work is not designed to compete with or to replace existing generic search engines. Rather, the CSDS in this work employs existing search engines as its initial searching mechanism for gathering information about the websites of Cloud services. Then, by consulting a Cloud ontology, the CSDS attempts to recognize an appropriate Cloud service among a list of several services. When a consumer submit requests to find Cloud services with specific requirements, the CSDS returns the best service and recommends other services for the user.

The objectives of this project are (1) to develop a CSDS (Sect. 2), (2) to design and construct a Cloud ontology (Sect. 4), and (3) a Cloud service reasoning agent (Sect. 3) for reasoning about the relations among Cloud concepts by consulting the Cloud ontology.

2 A Cloud Service Discovery System

This section illustrates the prototype of a Cloud service discovery system (CSDS) consisting of a search engine and three different agents, query processing agent, filtering agent, and Cloud service reasoning agent (CSRA). In Fig. 1, there are two main components, (1) a *CSDS*, which helps to find the best Cloud service in behalf of users, and (2) a *Cloud ontology*, which consists of taxonomy of concepts of

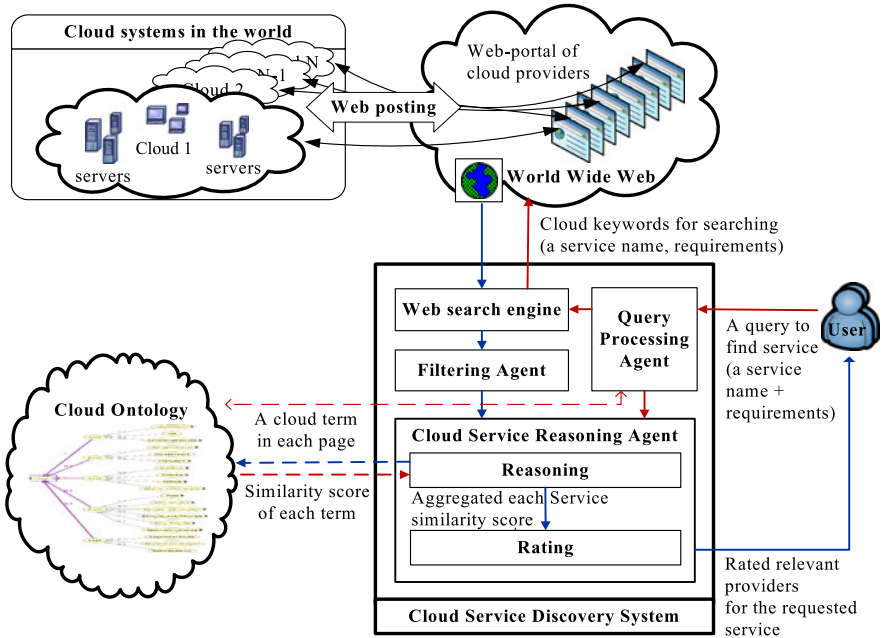


Fig. 1 Cloud service discovery system

different Cloud services to consult with the CSRA. In addition, there is a *user interface* that allows the user to enter queries containing a service name and requirements considered by their preferences.

Query Processing Agent (QPA). The QPA locates information sources by executing conventional search engines. Although the selection of a search engine is arbitrary, the default search engine is *Google (Search API)*. If the number of searched results is fewer than that specified by a user, generate new alternate queries to have more results [4]. The QPA works as follows:

1. Search with an exact service name (QPA refers to the Cloud ontology)
2. If the number of searched results ≤ 50 then,
 Generate new alternate queries used by its sibling nodes in Cloud ontology
 // With a different service name, but the service provides same (similar) functionalities

Filtering Agent. It is to relieve users of time consuming and laborious tasks of surfing many websites during an information retrieval process [4]. The filtering agent removes web-pages, (1) which cannot open URLs (not available) and (2) that sum of distance is high between keywords of users and a web-page by calculating distance of each term in the Cloud ontology.

Cloud Service Reasoning Agent (CSRA). It consults the Cloud ontology to reason about the relations among Cloud services. There are three reasoning methods to determine the similarity among services. Details of functionalities of the CSRA are given in Sect. 3.

3 A Cloud Service Reasoning Agent

A CSRA carries out two functions, (1) reasoning and (2) rating, as shown in Algorithm 1.

Reasoning. Reasoning is needed to understand a service and its similarity among a list of services. It makes a difference of the service similarity by calculating the similarity between the users input and services. A CSRA consults a Cloud ontology for performing service reasoning. All information supplied by a user is used to determine the similarity between two services. There are three methods to determine similarity, (1) similarity reasoning, (2) equivalent reasoning, and (3) numerical reasoning.

Rating. An aggregated similarity of each term in a web-page (i.e., service utility) is used to determine the rating. A web-page that has the highest service utility would be selected as the best service for the user. Other recommended services would be selected as well.

Algorithm 1

```

For all filtered web-pages  $\{Ft(1), Ft(2), \dots, Ft(N)\}$ 
  For number of all terms in a web-page,  $i = 1, 2, 3, \dots, n$ 
    1. Calculate similarity between  $q(i)$  in users queries  $\{q(1), q(2), \dots, q(n)\}$ ,
       and term  $t(i)$  in  $Ft(N)$   $\{t(1), t(2), \dots, t(n)\}$ .
       If two terms,  $q(i)$  and  $t(i)$ , are numeric values,
         (3) Numerical reasoning
       else
         (1) Similarity reasoning
       If two terms,  $q(i)$  and  $t(i)$ , are sibling nodes, then
         (2) Equivalent reasoning
    endFor
    2. Aggregate the similarity of all terms as ServiceUtility in the web-page
        $\{t(1), t(2), \dots, t(n)\}$ 
       [Aggregation method]
        $ServiceUtility = \sum_{k=1}^n term(k) \times weight(k)$ 
       where  $weight(k) = 1/n$  is uniformly distributed.
    endFor
    3. Rate a list of web-pages used by ServiceUtility in descending order.
    4. Select a web-page that has the highest ServiceUtility as the best Cloud
       service, and select other recommendation services as well.

```

4 Cloud Ontology

An ontology can provide meta information, which describes data semantics [5]. It provides a shared understanding of a domain of interest to support communication among human and computer agents [6]. An ontology contains a set of concepts and relationship between concepts, and can be applied to information retrieval to deal with user queries [7].

In Cloud computing, Clouds are generally divided into three different levels (*IaaS*, *PaaS*, and *SaaS* [8]).

Infrastructure as a Service (*IaaS*) [8] provides hardware, software, and equipments to deliver software application environments with a resource-usage-based pricing model.

Platform as a Service (*PaaS*) [8] offers a high-level integrated environment to build, test, and deploy custom applications. Generally, developers will need to accept some restrictions on the type of software they can write in exchange for built-in application scalability.

Software as a Service (*SaaS*) [8] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model.

The Cloud ontology in this work represents the relations among Cloud services to facilitate the CSRA in reasoning about the relations among Cloud service concepts (Figs. 2 and 3). It consists of 424 concepts constructed for the service reasoning. These include concepts of Cloud services that are currently being used and many services that may be released in the near future. There are three kinds of reasoning methods, (1) similarity reasoning, (2) equivalent reasoning, and (3) numerical reasoning.

4.1 Similarity Reasoning

Similarity reasoning is to calculate similarity between two concepts by counting common reachable nodes. The similarity of concepts represents the degree of commonality between concepts. We compute the semantic similarity based on the method in [10] as follows:

$$sim(x, y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1 - \rho) \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|} \quad (1)$$

where $\rho \in [0, 1]$ determines the degree of influence of generalizations.

$\alpha(x)$ is the set of nodes (upwards) reachable from x , we have $\alpha(x) \cap \alpha(y)$ as the reachable nodes shared by x and y , which is an indication of the commonality between concepts x and y [10].

For example, see Fig. 2 (*IaaS-InfraSoftware-OS*). In terms of Eq. 1, the concepts UNIX and Windows have 4 reachable nodes (upwards) from themselves, namely,

$\alpha(Unix) = 4$, $\alpha(Windows) = 4$, $\alpha(Linux) = 5$, $\alpha(Unix) \cap \alpha(Windows) = 3$, $\alpha(Unix) \cap \alpha(Linux) = 4$. Then, the similarity of $\alpha(Unix) \cap \alpha(Linux) = 4$ is greater than $\alpha(Unix) \cap \alpha(Windows) = 3$.

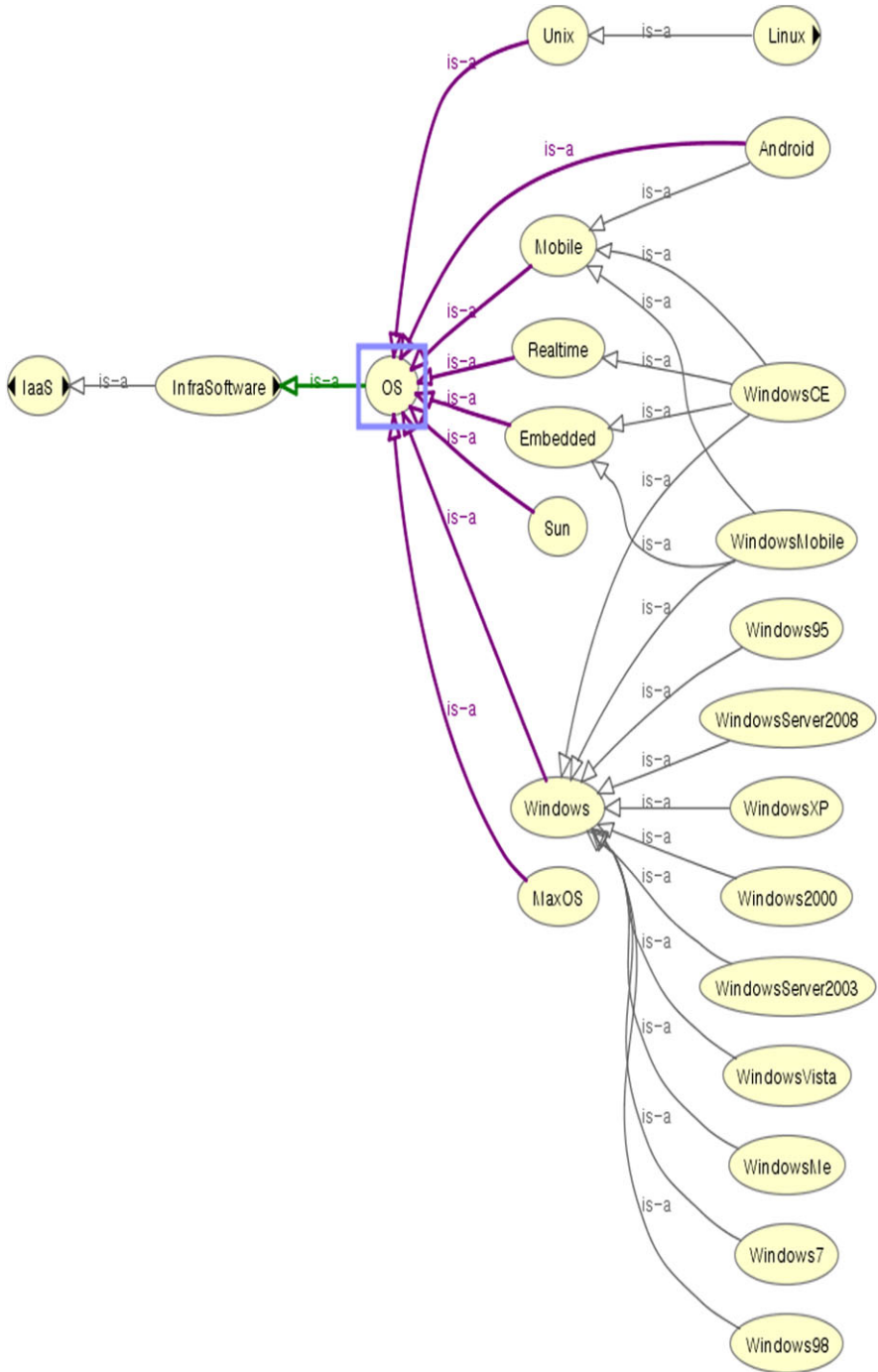
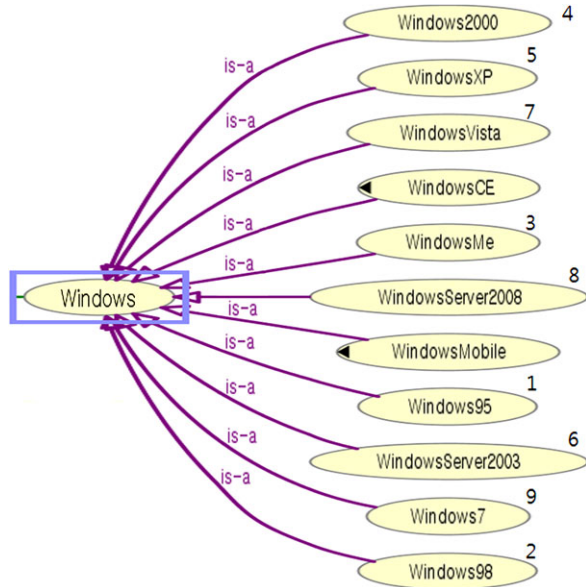


Fig. 2 Relation in terms of OS: IaaS

Fig. 3 Relation in terms of Windows



4.2 Equivalent Reasoning

Equivalent reasoning is used to calculate the similarity between two sibling concepts based on its labeled value. Since two sibling concepts have the same similarity, equivalent reasoning is designed to make the difference in the similarity between the concepts. We compute the similarity between two concepts, x and y , using the following method:

$$\hat{sim}(x, y) = sim(x, y) + \frac{(0.8^{|C_1 - C_2|})}{10} \tag{2}$$

where C_1 is a labeled value of concept x , C_2 is a labeled value of concept y , and $sim(x, y)$ is calculated by (1) similarity reasoning.

The values from 1 to 9 represent the chronological order of Windows (Fig. 3). This can be the way of classification of sibling concepts. In this way, “Windows95” is assigned the value 1 meaning the oldest version, and “Windows7” is assigned the value 9 meaning the latest version among all children nodes. Examples of showing the difference of the similarity between two sibling concepts are as follows.

a) Windows7 and WindowsVista

$$\frac{0.8^{|Windows7 - WindowsVista|}}{10} = \frac{0.8^2}{10} = 0.064$$

b) Windows7 and WindowsXP

$$\frac{0.8^{|Windows7 - WindowsXP|}}{10} = \frac{0.8^4}{10} = 0.041$$

Table 1 Maximum and minimum values of each concept

Concept	Max	Min
CPU clock	0.1	5.0
RAM	0.256	32
HDD	0.1	10000
Network bandwidth	0.1	10
Network latency	1	50000

Thus, even though two concepts are sibling nodes and could have same similarity, we make the even small difference between the concepts by equivalent reasoning.

4.3 Numerical Reasoning

Numerical reasoning is used to determine the similarity between two numeric values of concepts. We compute the similarity over numerical values using the following method.

$$sim(a, b, c) = 1 - \left| \frac{a - b}{Max_C - Min_C} \right| \quad (3)$$

where a and b are two numeric values and c is a name of concept that refers to maximum and minimum values of each concept in the Table 1.

For example, the similarity between two numeric values, 3.2 and 2.6, in the concept of CPU clock is determined as follows:

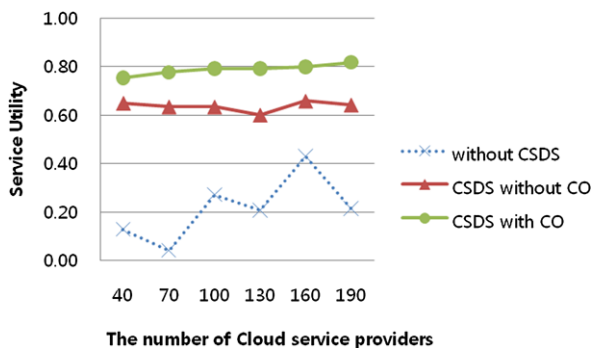
$$sim(3.2, 2.6, CPU\text{Clock}) = 1 - \left| \frac{3.2 - 2.6}{5.0 - 0.1} \right| = 0.88$$

5 Experimentation and Evaluation

The performance measures are (i) service utility (Fig. 4) and (ii) success rate (Fig. 6), with three comparison schemes, searching a Cloud service (1) without the CSDS, (2) the CSDS without the Cloud ontology, and (3) the CSDS with the Cloud ontology. In case (1), web-pages are searched with an exact service name and a web-page is randomly selected from the searched results. If it is a web-page about a Cloud service, then its service utility is determined. If not, the service utility is assigned as zero, which means that the discovery has failed. In case (2), web-pages that do not include a Cloud term are filtered out from the searched results, and a web-page is selected randomly from the filtered results, and the service utility is calculated. In case (3), web-pages are rated by the aggregated service utility, which is a result of the service reasoning. We conducted additional experimentation (Figs. 5 and 7) extended from [11] with three different compositions of reasoning methods,

Table 2 Experiment settings for simulations

Experiment variables	Value (range)
The number of providers	40, 70, 100, 130, 160, 190
The number of Cloud services provided by each provider	25 ~ 35(web-pages)
The number of Cloud service web-pages in the virtual-www	1200, 2100, 3000, 3900, 4800, 5700 (web-pages)
The number of web-pages in the virtual-www (not for Cloud service)	10000 web-pages
Total number of web-pages in the virtual-www	11200, 12100, 13000, 13900, 14800, 15700
The number of Cloud services	Around 100 service names
CPU clock	0.1 ~ 6.0 GHz
RAM size	0.256 ~ 36.0 GB
HDD size	0.1 ~ 1000 GB
Network bandwidth	0.1 ~ 10 Gbps
Network latency	1 ~ 5000 ms

Fig. 4 Service utility

(1) Sim. + Equ. + Num., (2) Sim. + Equ., and (3) Sim., to show how each reasoning method affects to the performance of the CSDS and the importance of all three reasoning methods for the CSDS as a result.

For evaluation purpose, we assumed that the WWW is replaced by the virtual-www for ease of testing. There are already 10000 web-pages (not for Cloud services) in a directory called the virtual-www, and around 30 web-pages are automatically generated by each provider when the CSDS is deployed. Depending on the number of providers generated, the total number of web-pages (i.e., Cloud services) would be decided between 11200 and 15700 in the virtual-www. The CSDS requires information containing a service name, the OS, CPU name, and the value of CPU clock, RAM, HDD, network bandwidth, and network latency (see Table 2).

Service Utility. In Fig. 4, the result of the CSDS with the Cloud ontology shows higher performance than without the CSDS and the CSDS without the Cloud on-

Fig. 5 Service utility:
Reasoning methods

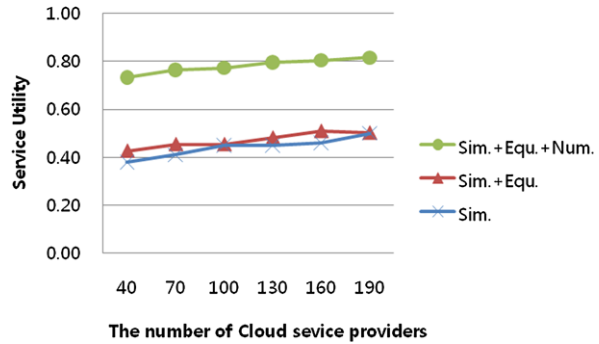


Fig. 6 Success rate

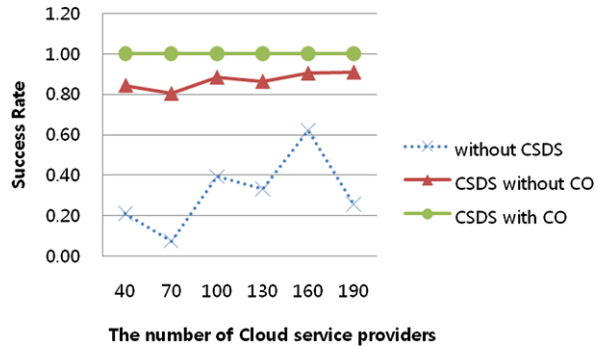
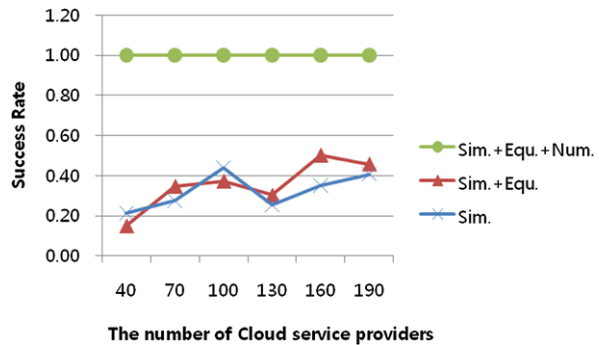


Fig. 7 Success rate:
Reasoning methods



tology in terms of the service utility. This is because the CSDS has filtering and reasoning functionalities, which means that web-pages of the Cloud service have higher chance to be selected and is more likely to be closer to users' requirements. In the Fig. 5, the result shows that selecting either without numerical reasoning or without equivalent reasoning and numerical reasoning may affect the performance of the CSDS in terms of service utility, since five numeric values (the value of CPU clock, RAM, HDD, network bandwidth, and network latency) should be considered for reasoning between users and Cloud services. The reasoning with similarity reasoning and equivalent reasoning methods show slightly better performance than

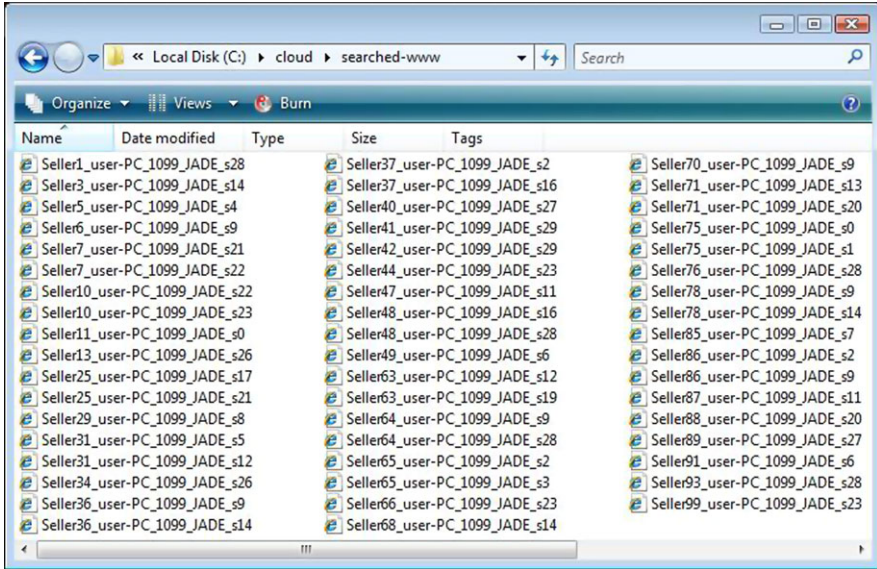


Fig. 8 After searching and filtering from the virtual-www

with only similarity reasoning, because the equivalent reasoning was designed to make a difference of the similarity between two sibling concepts (two sibling concepts have the same similarity by the similarity reasoning), and the value of similarity is very small calculated by equivalent reasoning (shown in Sect. 4.2, examples of equivalent reasoning).

Success Rate. In Fig. 6, success rate is calculated by the number of successes/ the number of attempts. It is assumed that a discovery will fail if the service utility is less than 0.5. Using the CSDS with the Cloud ontology, experimental results show that the service utility of retrieved web-pages is well over 0.5. The results demonstrated that using the CSDS with the Cloud ontology, users are more successful in discovering Cloud services. In Fig. 7, the result show that calculating the similarity between users and a Cloud service without (numerical) reasoning for numeric values is hard to over 0.5 in terms of service utility in both cases.

6 Proof-of-Concept Examples

An example is given in this section to demonstrate functionalities of the CSDS.

Step 0: Initially, when the CSDS is deployed, a number of Cloud providers are generated while each provider is posting around 25 of their services. A total of around 13000 web-pages existed in the virtual-www, including general web-pages.

Step 1: The screen in Fig. 10 shows the user input query, which contains a service name (e.g., “Visual_Studio_2010”) and requirements (e.g., OS = “Windows7,” CPU name = “Core2Quad,” CPU clock = “4.6,” RAM = “9.0,” HDD = “500.0,”

The screenshot shows a web browser window with the address bar containing 'C:\cloud\searched-www\Seller71_user-PC_1099_JADE_s13.html'. The page header includes a 'Customer Login' button and navigation links for 'Products', 'Services', 'Events', 'Community', and 'About'. The main content is a table with the following data:

cloud	(agent-identifier :name Seller71@user-PC:1099/JADE :addresses (sequence http://user-13 PC:7778/acc))	
Service Name	Visual_Studio_2010	
OS	WindowsVista	
CPU name	Core2Duo	
CPU Speed	5.0	GHz
Memory	10.0	GB
StorageSpace	700.0	GB
NetworkBandwidth	7.0	Gbps
NetworkLatency	2522.0	ms

Fig. 9 The best service to be founded

network bandwidth = “5.92,” network latency = “1667.0”), and mobile device support = “No”).

Step 2: The CSDS automatically searches with the exact service name “Visual_Studio_2010” from the virtual-www and filters out web-pages that do not include the “Cloud” term. The result is shown in Fig. 8.

Step 3: The CSDS consults the Cloud ontology for service reasoning. Then, the similarity of each term is aggregated as the service utility.

Step 4: The CSDS takes the highest utility, “0.8275,” as the best service among 53 web-pages and rate ordering.

Step 5: The CSDS returns the best service (e.g., provided by “Seller71”), which is the result of the service discovery as shown in Fig. 9. Additionally, results for the three cases, (1) without the CSDS, (2) the CSDS without the Cloud ontology, and (3) the CSDS with the Cloud ontology, are printed to the user interface screen. Other recommended services are also included in turn, as shown in Fig. 10.

7 Conclusion and Future Work

This paper has presented a Cloud service discovery system (CSDS). It is specially designed for users who want to find a Cloud service over the internet. A Cloud ontology is also introduced for enhancing the performance of the CSDS. The contributions of this work include (1) building of the CSDS and (2) constructing the

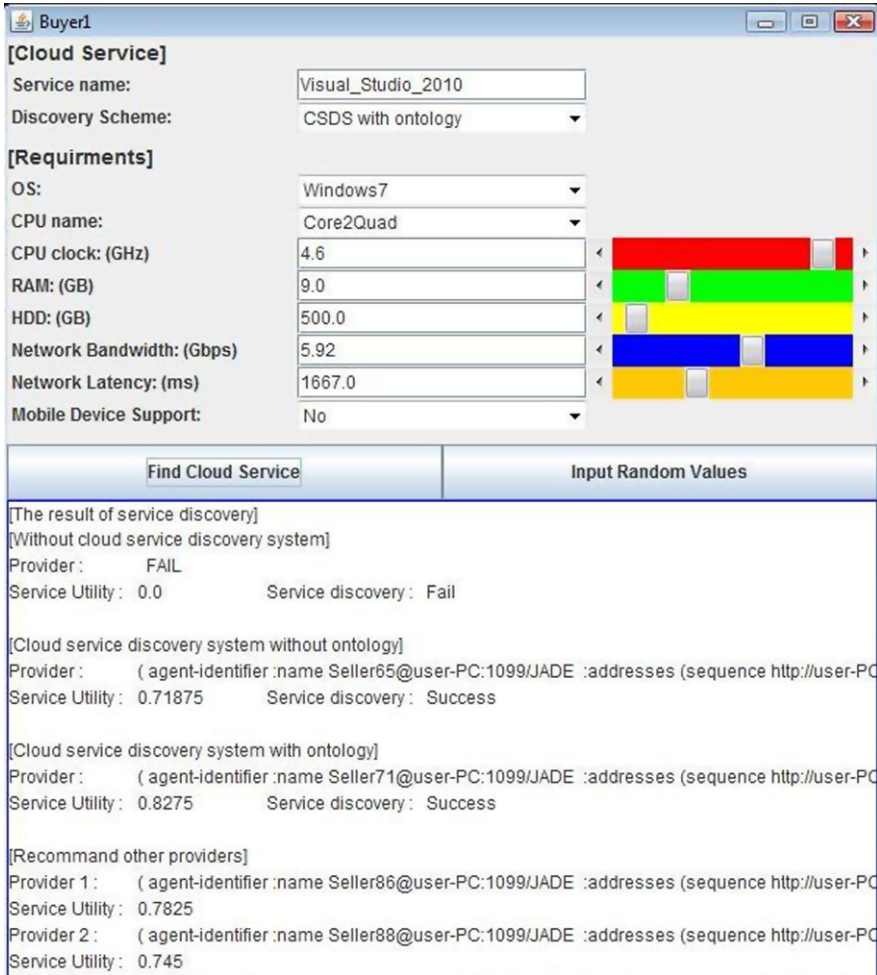


Fig. 10 Results of Cloud service discovery

Cloud ontology. It is the first attempt in building an agent-based discovery system that consults an ontology when retrieving information about Cloud services. In present, there are few big Cloud service providers and no various services. When the Cloud computing will be more commonly and widely used in the near future, it can be helpful for Cloud users who want to find a Cloud service under their specific preference.

From the empirical results in Sect. 5, the CSDS with the Cloud ontology achieved better performance than the CSDS without the Cloud ontology. By consulting a Cloud ontology to reason about the relations among Cloud services, the CSDS is more successful in locating Cloud services and more likely to discover Cloud services that meet users requirements. In addition, the CSDS essentially needs all three

kinds of reasoning methods to have high performance in terms of service utility and success rate. Since this work is a preliminary work for Cloud service discovery, a list of future agendas includes the following. (1) Designing of Cloud (service) ontology is still an issue that is not fully addressed. (2) Although Cloud services and traditional web service are currently described by Web Services Description Language (WSDL) [12], we had to design providers' services in our own model because of the lack of standards and experimentation.

Acknowledgements This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092) and the DASAN International Faculty Fund (project code: 140316).

References

1. Birman, K., Chockler, G., van Renesse, R.: Toward a Cloud computing research agenda. In: Cloud Computing Defined
2. Youseff, L., Butrico, M., Da Silva, D.: Toward a unified ontology of cloud computing. In: Grid Computing Environments Workshop, GCE 2008 (2008)
3. Sheu, P.C.-Y., Wang, S., Wang, Q., Hao, K., Paul, R.: Semantic computing, Cloud computing, and semantic search engine. In: 2009 IEEE International Conference on Semantic Computing
4. Sim, K.M., Wong, P.T.: Toward agency and ontology for web-based information retrieval. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **34**(3), 1–13 (2004)
5. Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Berlin (2003)
6. Stuckenschmidt, H.: *Ontology-based information sharing in weekly structured environments*. Ph.D. thesis, AI Department, Vrije University, Amsterdam (2002)
7. FIPA 2001. Foundation for intelligent physical agents: FIPA Ontology Service Specification. <http://www.fipa.org/specs/fipa00086/XC00086D.html>
8. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: IEEE Grid Computing Environments (GCE 2008), Texas, pp. 1–10 (2008)
9. Andreasen, T., Bulskov, H., Kanppe, R.: From ontology over similarity to query evaluation. In: 2nd International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information System (ODBASE), Catania, Sicily, Italy, 3–7 November 2003
10. Han, L., Berry, D.: Semantic-supported and agent-based decentralized grid resource discovery. *Future Gener. Comput. Syst.* **24**, 806–812 (2008)
11. Han, T., Sim, K.M.: An ontology-enhanced Cloud service discovery system. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 644–649 (2010)
12. Zeng, C., Guo, X., Ou, W., Han, D.: Cloud computing service composition and search based on semantic. In: *CloudCom 2009. LNCS, vol. 5931*, pp. 290–300 (2009)

Possible Applications of Navigation Tools in Tilings of Hyperbolic Spaces

Maurice Margenstern

Abstract This paper introduces a method of navigation in a large family of tilings of the hyperbolic plane and looks at the question of possible applications in the light of the few ones which were already obtained. (This paper is a revised and slightly extended version of a paper presented by the author at IMECS'2010, see Margenstern (Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, pp. 367–382, 2010).)

Keywords Hyperbolic tilings · Cellular automata · Applications

1 Introduction

Hyperbolic geometry appeared in the first half of the 19th century, proving the independence of the parallel axiom of Euclidean geometry. Models were devised in the second half of the 19th century. Here, we use one of the most popular ones, Poincaré's disc. This model is represented by Fig. 1.

From a famous theorem established by Poincaré in the late 19th century, it is known that there are infinitely many tilings in the hyperbolic plane, each one generated by the reflection of a polygon P in its sides and, recursively, in the reflection of the images in their sides, provided that the number p of sides of P and the number q of copies of P which can be put around a point A and exactly covering a neighbourhood of A without overlapping satisfy the relation: $\frac{1}{p} + \frac{1}{q} < \frac{1}{r^2}$. The numbers p and q characterize the tiling which is denoted $\{p, q\}$ and the condition says that the considered polygons live in the hyperbolic plane. Note that the three tilings of the Euclidean plane which can be defined up to similarities can be characterized by the relation obtained by replacing $<$ with $=$ in the above expression. We get, in this

M. Margenstern (✉)
Université Paul Verlaine-Metz, LITA, EA 3097, UFR-MIM, Campus du Saulcy,
57045 Metz Cédex, France
e-mail: margens@univ-metz.fr

Fig. 1 Poincaré's disc model: inside the open disc, the points of the hyperbolic plane. Lines are trace of diameters or circles orthogonal to the border of the disc, e.g. the line m . Through the point M we can see a line s which cuts m , two lines which are parallel to m : p and q , touching m in the model at P and Q respectively which are points of the border and are therefore called points at infinity. At last, and not the least: the line n also passes through M without cutting m , neither inside the disc nor outside it

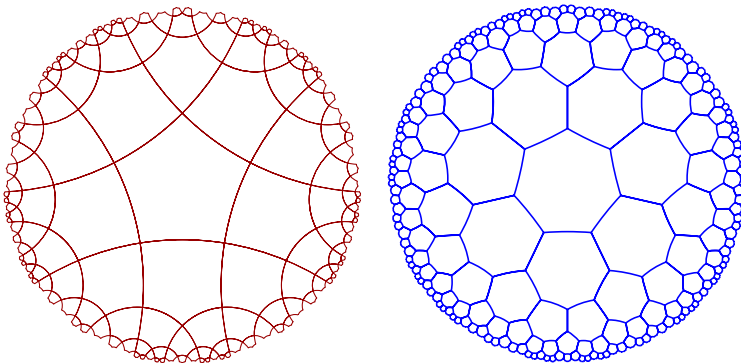
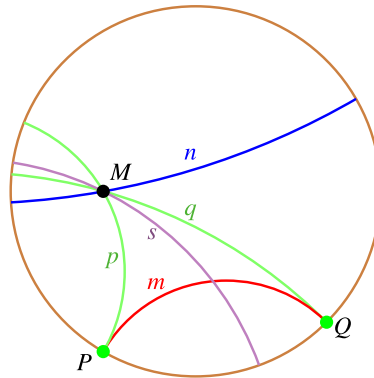


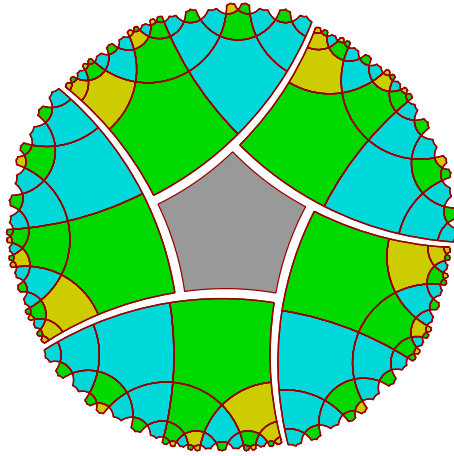
Fig. 2 Left-hand side: the pentagrid; right-hand side: the heptagrid

way, $\{4, 4\}$ for the square, $\{3, 6\}$ for the equilateral triangle and $\{6, 3\}$ for the regular hexagon.

In the paper, we shall focus our attention on the simplest tilings which can be defined in this way in the hyperbolic plane: $\{5, 4\}$ and $\{7, 3\}$. We call them the **pentagrid** and the **heptagrid** respectively, see Fig. 2.

To navigate in these tilings was for a long time a non-trivial question. In 1999 and 2000, see [4, 14], the author found a technique which allows to find one's way in these tilings, first in the pentagrid. A bit later, the technique was generalized to the heptagrid and to infinitely many other tilings of the hyperbolic plane. Details and references on these results can be found in [9, 10] as well as their extension to one tiling of the hyperbolic 3D space and another one of the hyperbolic 4D space.

Fig. 3 First part of the splitting: around a central tile, fixed in advance, five sectors. Each of them is spanned by the Fibonacci tree defined in Fig. 4



Cellular automata are a tool used in various sciences, from gas statistical physics to economy, for simulation purposes with good results and a few industrial applications. We refer the reader to proceedings of the last two issues of **ACRI** conferences to have a look at this range of applications.

The navigation technique introduced in [4] allowed to implement cellular automata in the pentagrid and in the heptagrid and to devise a few applications.

In Sect. 2, we sketchily describe the navigation technique. In Sect. 3, we remind the results on cellular automata and in Sect. 4, we consider the applications already performed as well as a few others which could be useful. In Sect. 5, we conclude with what could be done in future work.

2 Navigation in the Pentagrid and in the Heptagrid

The navigation in a tiling of the hyperbolic space can be compared to the flight of a plane with instruments only. Indeed, we are in the same situation as a pilot in this image as long as the representation of the hyperbolic plane in the Euclidean one entails such a distortion that only a very limited part of the hyperbolic plane is actually visible.

The principle of the navigation algorithms rely on two ideas which we illustrate on the pentagrid. The first one, is a way to split the hyperbolic plane, see Fig. 3 and Fig. 4: the recursive structure of this splitting defines a tree which spans the tiling. The second idea consists in numbering the nodes of the tree level after level and, remarking that the number of nodes on the level n is f_{2n+1} defined by the Fibonacci sequence where $f_0 = f_1 = 1$, to represent the numbers in the numbering basis defined by this sequence. Also, as the representation is not unique, we fix it by choosing the longest one.

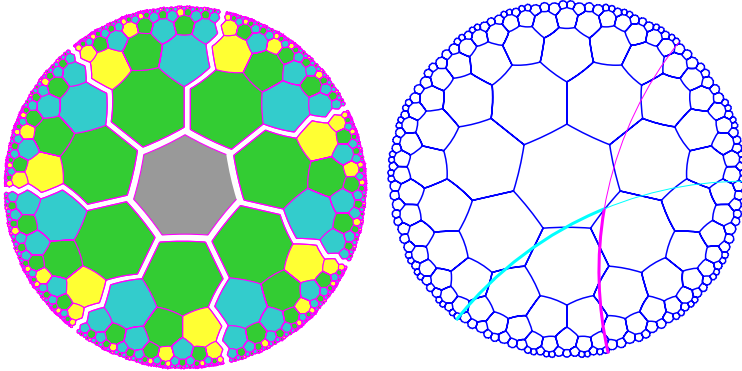
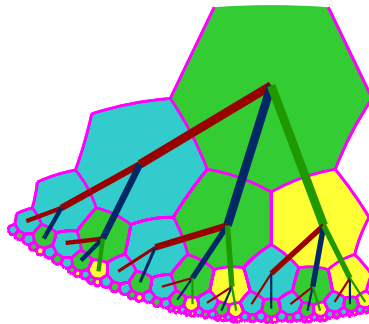


Fig. 6 *Left-hand side*: first part of the splitting, around a central tile, fixed in advance, seven sectors. Each of them is spanned by the Fibonacci tree defined in Fig. 4, see the right-hand side picture. *Right-hand side*: the mid-point lines. This tool which shows how a sector spanned by the Fibonacci tree is defined in the heptagrid

Fig. 7 The heptagrid is spanned by the same tree as the pentagrid: compare with the tree of Fig. 4



From the preferred son properties, it was possible to devise an algorithm which computes the path from the root to a node in a linear time in the length of the coordinate of the node, see [5]. From this, we also get that the coordinates of the neighbours of a given tile can be computed from the coordinate of the tile in linear time too.

It is interesting to remark that the heptagrid possesses properties which are very similar to those of the pentagrid, see Figs. 6 and 7. Note that the right-hand side picture of Fig. 6 explains why the same tree basically spans each of the seven sectors of the left-hand side of the figure.

We conclude this section by mentioning that these nice properties of the pentagrid and of the heptagrid can be extended to two infinite families of tilings of the hyperbolic plane. In particular, the fact that the same tree spans the pentagrid and the heptagrid can be extended as follows: for each p , $p \geq 5$, the same tree spans the tiling $\{p, 4\}$ and the tiling $\{p + 2, 3\}$. The trees are different for different values of p although they share a common feature: nodes are divided into black and white

nodes. The difference in the number of sons from a white node to a black one is 1. The unique black son can be chosen to be the leftmost one.

3 Results on Cellular Automata in Hyperbolic Spaces

These fast algorithms allowed to implement cellular automata in the hyperbolic plane, first in the pentagrid and, a few years later, in the heptagrid.

We have three kinds of results regarding cellular automata in these spaces: complexity results, universality results and a solution of two problems dealing with communications between cells of a cellular automaton. Although these results have a definite theoretical character, they nevertheless have a practical significance.

The most striking result regarding complexity, is that for cellular automata in the hyperbolic plane, we have $\mathbf{P}_{hc} = \mathbf{NP}_{hc}$. This means that non-deterministic polynomial time computations of a cellular automaton in the hyperbolic plane can be performed also in polynomial time by a deterministic cellular automaton in the hyperbolic plane. Moreover, the exact power of computation of the class \mathbf{P}_{hc} is the well-known class \mathbf{PSPACE} . The reader is referred to [2, 3, 10, 14].

For what is universality, there are universal cellular automata in the hyperbolic plane, *i.e.* able to simulate the computation of any Turing machine. What is more important is that if infinite but elementary initial configurations are allowed – within this limited room we cannot formally describe the exact meaning of this expression – it is possible to simulate any Turing machine with a cellular automaton with 9 states in the case of the pentagrid, see [16], and with 6 states in the case of the heptagrid, see [15] and even less: 4 states, see [12]. All the quoted cellular automata have a common structure: they simulate a railway circuit traversed by a unique locomotive consisting of tracks and three kinds of switches devised in [19], see also [10].

Now, the linear algorithm for finding the path from a node to the root of the sector to which the node belongs allows to devise efficient communication protocols for cells of a cellular automaton in the pentagrid or in the heptagrid, see [7, 8, 10].

If the latter point is closer to applications, the first ones also say something on this regard. The meaning of the complexity results is that hyperbolic cellular automata may run much faster than their Euclidean analogues: they have at their disposal an exponential area which can be constructed and used in linear time. Also, these computations may be universal as this is the case for their Euclidean analogues. And so, we can do any computations in this frame, never in more time than what is required for a Euclidean cellular automaton, and very often in much less time. In particular, in this setting, hard problems of everyday life turn out to be solvable in polynomial time, very often even in linear time.

4 On the Side of Applications

The above results might seem too beautiful to be true. However, they are theoretical results whose proofs were checked, some of them with the help of a computer program, and they are correct.

But is this feasible?

Our local environment is usually thought as Euclidean although it would be better to see it as governed by spherical geometry. Many cosmologists consider our universe as a space with a negative curvature. In this regard, hyperbolic geometry could be a better model for medium scale than Euclidean geometry. Computations on the orbit of Mercury are more conformal to observations when performed in a hyperbolic setting. So, our tools might have applications at this scale which is not a today urgent matter, but we know that possible tools exist.

I would also mention another argument. An important feature of hyperbolic geometry is the lack of similarity. As a consequence, it can be said that a shape has a certain size necessarily. As an example, in the hyperbolic plane, there is a unique size of the edge for a regular pentagon with right angles. This means that two such pentagons can be transformed from each other through a simple geometric transformation which is an isometry. And so, up to isometries, such a pentagon is unique. This property is shared by any figure of the hyperbolic plane. Now, if we look at biology, we can notice that there is no true similarity. Individual differences, for instance, are not addressed by similarity. This remark led me to [6] where I used the pentagrid to represent a theoretical model of living cells, the common point being the fact that, in both cases, a tree underlies the structures.

Now, another field of application, more important in my eye, is provided by computer science itself. We need new concepts to handle problems raised by massive computations and by the management of huge amounts of information. For this, we need new horizons and it is not at all unreasonable to consider tilings of the hyperbolic plane with their navigation tools as a possible model for tackling these problems. Note, for instance, that trees are already used in the organisation of operating systems: many a user is faced with the tree structure of the directories of his/her machine. Now, trees are also spanning hyperbolic geometry and this is particularly blatant in the field of tilings in the hyperbolic plane as I hope the reader is convinced after reading Sect. 2.

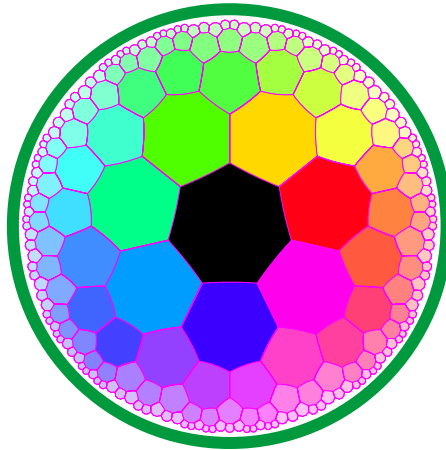
In the present section, we first have a look at the existing applications, see the next two subsections. These applications are based on a **fisher eye** effect of the pentagrid and the heptagrid. Now, it is known that fisher eye techniques are of interest for several applications in human-machine interaction. In our last subsection, we again address the problem of representing, storing and exploring information which mainly use the coordinate system.

4.1 The Colour Chooser

The colour chooser is a software which was developed in my laboratory by members of my team, see [1, 10]. From the initial state of the chooser illustrated by Fig. 8, the user can navigate in the tiling in order to look after the colour he/she thinks the most adapted for his/her purpose.

There are seven fixed in advance keys which indicate to which tile the user wishes to go from the central tile. Once the appropriate key is pressed, the chosen tile comes

Fig. 8 (Color online) The colour chooser: the pentagrid is also a possible tool but the heptagrid gives the best representation. Probably because at first glance, the heptagons of the figure are seen as hexagons: it is needed to count the sides in order to detect the difference



to the centre of the disc and all the other tiles move correspondingly. Consequently, the navigation appears as if the green disc of Fig. 8 would be a window moving over the hyperbolic plane. Once the black tile indicating the initial centre is no more visible, it is very easy to get lost. To avoid such a defect, the chooser keeps an arrow pointing at a point of the green disc to which the user have to go in order to go back to the initial centre.

4.2 The Keyboards

Another application was developed in the laboratory, see [18], which we consider in the next sub-subsection and which was developed in a different direction, as will be seen in the other sub-subsection of this subsection.

4.2.1 Latin Keyboard

The first idea was a proposal of a keyboard for cell phones. As our laboratory is in France, tests were performed with French students on randomly chosen sequences of French sentences devised for the experiment.

Figure 9 illustrates the basic principle of the software. The idea is close to that of the chooser. But this time, the tiles contain letters and the user goes from the initial empty centre to the desired letter by pressing appropriate keys. At most three keys have to be pressed and with other media, pressing the keys can be replaced by three quick moves of the hand.

Also, the pentagrid is used instead of the heptagrid because for this purpose it is more suited to the gestures of the user and letters can better be seen. The reason is that the regular pentagon with right angle is bigger than the rectangular heptagon with the angle $\frac{2\pi}{3}$. This explains that letter can better be seen. This explains

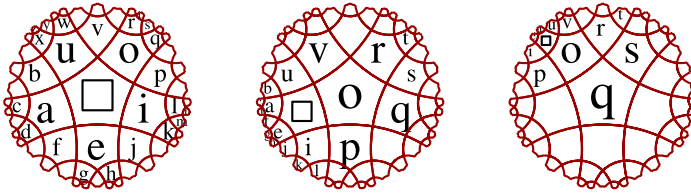
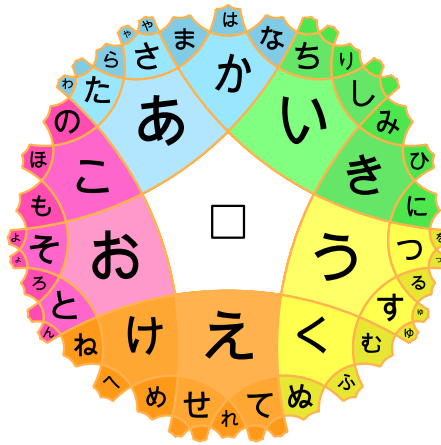


Fig. 9 A proposal of a keyboard for cell phones

Fig. 10 A proposal of a keyboard for Japanese cell phones



also that less accuracy is required from the user for his/her gestures. The experiments proved that the keyboard is certainly not worse than commercial products. Moreover, among young people, it raised a curiosity which contributed to the quick learning of the keyboard. Another important feature is that we adopted a distribution of the letters as close as possible to the standard alphabetic order, which requires no additional effort of memory from the user.

4.2.2 Japanese Keyboard

The use of the pentagrid and the fact that hiraganas and katakanas of the Japanese language are traditionally presented in series based on the five vowels of the language inspired me the idea to use the same grid but this time, in a Japanese environment. We did some work in this direction, with Japanese colleagues, see [17]. Figure 10 illustrates the principle of the working.

This keyboard has always met a big success in the conferences where I presented it in Japan. There was a prototypical implementation on actual cell phones and concluding experiments were performed with a small group of Japanese students on a protocol which was similar to the one followed in France but, of course, adapted to the Japanese language.

It is important to indicate that our project also aims at a full representation of the kanjis used in the Japanese language, starting from a phonetic approach. This is a very complex task and, at the present moment it is not completed.

The question could be raised of the adaptation of the same principle to other languages. I was indicated by several colleagues that the occurrence of exactly five vowels is a common feature of many Asian languages as the languages from Malaysia and Philippines, also including the Polynesian languages. It would be interesting to see whether similar ideas could be developed for other languages: any proposal would be welcomed!

4.3 *Other Possible Applications*

Now, let us turn to other possible applications which were not yet tested and which are based on other principles.

As indicated in the beginning of this section, computer science and computer engineering could be an important field of application of the navigation technique introduced by Sect. 2.

There is not enough room to develop such applications in a detailed way. This is why I shall mention three possible sub-fields of application and give general arguments only in favour of these proposals.

These three sub-fields are: the representation of the Internet, computer architecture and operating systems with data processing.

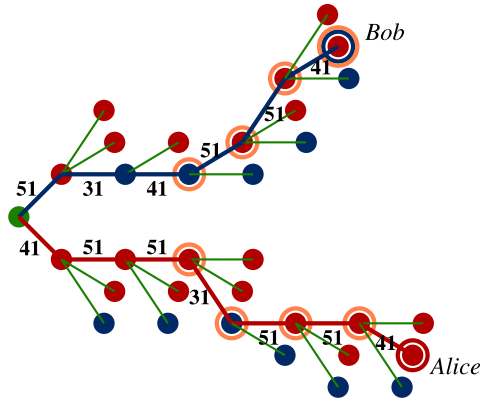
The Internet can be represented as a graph. This is often performed in force oriented representations which assign a mass to each node and define their respective positions by application of the laws of mechanics. Using the pentagrid or the heptagrid can be an alternative representation. In [10], I illustrate this point by defining addresses of nodes which is based on the navigation technique of Sect. 2 but which looks like the today used IP numbers. There is another advantage in this proposal. For both the pentagrid and the heptagrid, [10] gives an algorithm which, from the coordinates of two tiles A and B , gives a shortest path from A to B in linear time in the size of the coordinates. This is an important improvement of the path algorithm from a node to the root of the sector of the node mentioned in Sect. 2. It also allows to define a fast algorithm for changing the centre of coordinates.

This possibility may better represent the Internet connections by giving addresses which are continuous with respect to the connection distance between nodes.

4.3.1 **Communications Between Tiles**

In [8], I introduced a protocol of communication between tiles of the pentagrid or the heptagrid based on the following principle: there is an absolute system of coordinates but when sending a message, a tile considers itself as the central tile and sends a message to all tiles together with its **relative** address which is 0. Now each

Fig. 11 Illustration of the protocol of communication between tiles of the pentagrid or of the heptagrid. The circled tiles indicate a shortest path from Alice to Bob. The coordinates give both numbers of the edge shared by two tiles by giving the number of the side through which the other tile can be seen. The number of the side is defined with respect to that of the father in the tree which is 1. The other numbers are obtained by counter-clockwise turning around the tile



tile which receives the message, conveys it to its sons with respect to the coordinate system of the sender. It also appends an information of constant size in order to form the address it will convey to its sons. This allows a tile which receives the message and which wishes to establish a contact with the sender to send an answer to the appropriate sender: it simply reverses the address in a way which is described in [8]. Figure 11 illustrates the protocol on a toy example and gives the principle of the coordinate systems.

There, Alice, the inhabitant of a tile sends a message to everybody. At time 8, Bob, who lives in another cell receives the message and decides to give an answer. Together with the message, Bob receives his own address in Alice's system of coordinates. Accordingly, the reversal of the address gives Alice's address in Bob's own system of coordinates.

4.3.2 Internet, Processors and File Storage

This can be transported to a set of processors: they could be organised as if they would stand in a disc of the pentagrid or of the heptagrid around a central one which would not necessarily have a control function. It is enough for that to assign them coordinates as those introduced in Sect. 2. The communication between processors could be organised according to the just described protocol.

The shortest path algorithm can be used for the representation of the Internet but also for the representation of a file storage for an operating system. The system of coordinates which allows to construct a shortest path from one point to another in a linear time with respect to both coordinates should facilitate the finding of queries as topological neighbourhood is up to a point reflected in the coordinates themselves. From this remark, we can see that this can be of help also for data processing. In particular, constant saving mechanisms can be organised by scanning a circular area by a branch which moves from a central point to an indicated point of the circumference. The idea is that the branch constantly moves around this circumference and

that at each time, the content of the branch only is saved. After a certain time, everything is saved and updates can also be included in this constant saving provided that it does not exceed a fixed in advance amount. In case of exceeding the threshold, the update is split into as many parts as needed in order that each part should fall within the threshold.

It can be noted that when the branch goes from one node of the circumference to the next one, in many cases, the two positions of the branch may have a long common interval I starting from the centre which consists of the same nodes. If no update occurs on the nodes belonging to I between the two corresponding tops of saving, then only a small part of the new branch has to be saved. This can be easily managed by an appropriate signalization on the branch, see [11].

5 Conclusion

There is still much work to do in this domain, both in theory but also for applications.

Practical problems are difficult as this is witnessed by the example of scheduling problems, either in airplane traffic or in production processes: such problems are NP-complete. We have seen that the frame proposed by this paper allows to solve them in polynomial time. This is not the single theoretical approach leading to such results. As an example, molecular computations based on a modelling of DNA strand reactions or on a modelling of a living cell lead to similar results. But our frame has the advantage of not being concerned by still unsolved biological problems for a practical implementation, the nature of which is not yet well understood: either it comes from fundamental issues or it comes from not well enough mastered techniques. The last subsection of Sect. 4 pointed at fields where our approach might have feasible applications.

It seems to me that exchanging all possible ideas is a way to find out paths which will turn out to give the expected solution with, in many cases, surprising outcomes. This paper aims at being a contribution to this large exchange. I hope that the already few applications explored so far will be followed by many ones. I hope that it will encourage people to venture along the tracks opened in this paper and, it would be the best, to go further towards new avenues.

References

1. Chelghoum, K., Margenstern, M., Martin, B., Pecci, I.: Palette hyperbolique: un outil pour interagir avec des ensembles de données (Hyperbolic chooser: a tool to interact with data sets, in French). In: IHM'2004, Namur (2004)
2. Iwamoto, Ch., Margenstern, M.: Time and space complexity classes of hyperbolic cellular automata. IEICE Trans. Inform. Syst. **387-D**(3), 700–707 (2004)
3. Iwamoto, Ch., Margenstern, M., Morita, K., Worsch, Th.: Polynomial time cellular automata in the hyperbolic plane accept exactly the PSPACE languages. In: SCI'2002 (2002)
4. Margenstern, M.: New tools for cellular automata in the hyperbolic plane. J. Universal Comput. Sci. **6**(12), 1226–1252 (2000)

5. Margenstern, M.: Implementing cellular automata on the triangular grids of the hyperbolic plane for new simulation tools. In: *ASTC'2003*, Orlando, March 29–April 4 (2003)
6. Margenstern, M.: Can hyperbolic geometry be of help for P systems? *Lect. Notes Comput. Sci.* **2933**, 240–249 (2004)
7. Margenstern, M.: A new way to implement cellular automata on the penta- and heptagrids. *J. Cell. Autom.* **1**(1), 1–24 (2006)
8. Margenstern, M.: On the communication between cells of a cellular automaton on the penta- and heptagrids of the hyperbolic plane. *J. Cell. Autom.* **1**(3), 213–232 (2006)
9. Margenstern, M.: *Cellular Automata in Hyperbolic Spaces*, vol. 1: Theory. Old City Publishing, Philadelphia (2007), 422 p
10. Margenstern, M.: *Cellular Automata in Hyperbolic Spaces*, vol. 2: Theory. Old City Publishing, Philadelphia (2008), 360 p
11. Margenstern, M.: A uniform and intrinsic proof that there are universal cellular automata in hyperbolic spaces. *J. Cell. Autom.* **3**(2), 157–180 (2008)
12. Margenstern, M.: Universal cellular automata in hyperbolic spaces. In: *13th WSEAS International Conference on Computers*, Rodos, July 23–25, pp. 83–89 (2009), ISBN 978-960-474-099-4
13. Margenstern, M.: Navigation in tilings of the hyperbolic plane and possible applications. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, Hong Kong, 17–19 March 2010, pp. 376–382 (2010)
14. Margenstern, M., Morita, K.: A polynomial solution for 3-SAT in the space of cellular automata in the hyperbolic plane. *J. Universal Comput. Syst.* **5**, 563–573 (1999)
15. Margenstern, M., Song, Y.: A universal cellular automaton on the ternary heptagrid. *Electron. Notes Theoret. Comput. Sci.* **223**, 167–185 (2008)
16. Margenstern, M., Song, Y.: A new universal cellular automaton on the pentagrid. *Parallel Process. Lett.* **19**(2), 227–246 (2009)
17. Margenstern, M., Martin, B., Umeo, H., Yamano, S., Nishioka, K.: A proposal for a Japanese keyboard on cellular phones. *Lect. Notes Comput. Sci.* **5191**, 299–306 (2008)
18. Martin, B.: VirHKey: a VIRTUAL Hyperbolic KEYboard with gesture interaction and visual feedback for mobile devices. In: *Mobile HCI*, pp. 99–106 (2005)
19. Stewart, I.: A subway named Turing. *Mathematical recreations. Sci. Am.* 90–92 (1994)

Graph Pattern Matching with Expressive Outerplanar Graph Patterns

Hitoshi Yamasaki, Takashi Yamada,
and Takayoshi Shoudai

Abstract An outerplanar graph is a planar graph that can be embedded in the plane in such a way that all vertices lie on the outer boundary. Outerplanar graphs express many chemical compounds. An externally extensible outerplanar graph pattern (*eoo-graph pattern* for short) represents a graph pattern common to a finite set of outerplanar graphs, like a dataset of chemical compounds. The eoo-graph pattern can express a substructure common to blocks that appear in outerplanar graph structured data. In this paper, we propose a polynomial time algorithm for deciding whether or not a given eoo-graph pattern matches a given connected outerplanar graph. Moreover, we show the expressiveness of the pattern class by experiments on a chemical compound database.

Keywords Graph pattern matching · Graph mining · Outerplanar graphs · Chemical graphs

1 Introduction

Large amounts of data that have graph structures (such as map data, CAD, biomolecular, chemical molecules, and the World Wide Web) are stored in databases. For example, HTML/XML documents can be expressed by ordered trees, and almost all chemical compounds in the NCI dataset [6], which is a popular graph mining dataset, are expressed by outerplanar graphs. Outerplanar graphs are planar graphs embedded in the plane in such a way that all vertices lie on the outer boundary. In Fig. 1, graphs g_1 , g_2 , g_3 , G , and H are examples of outerplanar graphs. Many researchers are interested in acquiring knowledge from data that have structures

This is a revised and extended version of a paper [9] presented at the International MultiConference of Engineers and Computer Scientists, Hong Kong, 2010.

H. Yamasaki (✉)

Department of Informatics, Kyushu University, Fukuoka 819-0395, Japan
e-mail: h-yama@i.kyushu-u.ac.jp

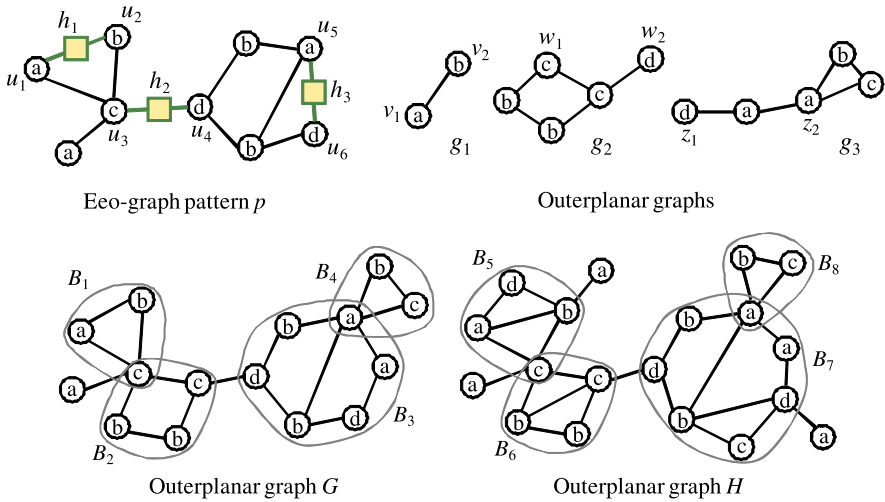


Fig. 1 An eeo-graph pattern p and outerplanar graphs $g_1, g_2, g_3, G,$ and H over a vertex label set $\{a, b, c, d\}$: A variable is drawn by a box with lines to its elements

such as sequences, trees, or graphs [1, 3, 8]. Horváth et al. [4] proposed a frequent subgraph mining algorithm for outerplanar graphs. For graph mining algorithms, graph pattern matching algorithms play a key role throughout the computations.

For graph G , we call a maximal biconnected subgraph of G a *block* of G if it has at least three vertices. For example, in Fig. 1, B_1, \dots, B_8 are blocks. A block of an outerplanar graph has a unique planar embedding up to the mirror image. Thus the edges of an outerplanar graph are classified into two types; *external* and *internal*. External edges lie on the external face, while internal edges do not. An external edge that does not belong to any block is called a *bridge*. Because an internal edge must belong to a block, it is also called a *diagonal* of the block. For an integer $d \geq 0$, an outerplanar graph is said to be d -tenuous if each of its blocks contains at most d diagonals. Horváth et al. [4] proposed an Apriori-like algorithm for enumerating all frequent d -tenuous outerplanar subgraphs in a finite set of outerplanar graphs. In [7], we introduced a graph-structured pattern, called a *block preserving outerplanar graph pattern* (*bpo-graph pattern* for short), which is an outerplanar graph with structured variables, and proposed a refinement-based technique for enumerating all maximal frequent bpo-graph patterns in a finite set of outerplanar graphs. A bpo-graph pattern represents an expressive graph structure among blocks. However it cannot represent internal structures common to different blocks if the blocks are not isomorphic.

Our final objective in this research is to propose an efficient data mining method for extracting more expressive outerplanar graph-structured patterns, which simultaneously represent connection patterns and internal structured patterns common to different blocks. In this paper, we introduce a graph pattern that has structured variables, called an *externally extensible outerplanar graph pattern* (*eeo-graph pattern*

for short). A variable of an eeo-graph pattern is a vertex pair that is not currently an edge in the graph and that becomes an external edge if added as an edge to the graph. Then a variable of an eeo-graph pattern is called an *external variable*. External variables can be replaced with any connected outerplanar graphs that satisfy certain conditions. In Fig. 1, a graph pattern p is an example of eeo-graph patterns, and an outerplanar graph G is obtained from p by removing variables h_1 , h_2 , and h_3 , and identifying vertex pairs (u_1, u_2) , (u_3, u_4) , and (u_5, u_6) of p with (v_1, v_2) of g_1 , (w_1, w_2) of g_2 , and (z_1, z_2) of g_3 , respectively. For an eeo-graph pattern p and a connected outerplanar graph G , we say that p matches G if there is such a variable replacement by which the graph obtained from p is subgraph isomorphic to G under the condition of preserving blocks. In this paper, we propose a polynomial time algorithm for the pattern matching problem for the class of eeo-graph patterns. Moreover, we show the expressiveness of the pattern class by experiments on a chemical compound database.

2 Externally Extensible Outerplanar Graph Patterns

Let $G = (V, E)$ be a graph. Let Λ , Δ , and X be infinite alphabets, where $(\Lambda \cup \Delta) \cap X = \emptyset$. A *variable* of G is a list of different vertices of V , which is denoted by (v_1, \dots, v_ℓ) ($\ell \geq 1$), where $v_i \in V$ ($1 \leq i \leq \ell$) and $v_i \neq v_j$ if $i \neq j$ ($1 \leq i, j \leq \ell$). All vertices and edges are labelled with symbols in Λ and Δ , respectively. In this paper, all variables are assumed to be labelled with mutually distinct symbols in X . Then a triple $p = (V, E, H)$ is called a *graph pattern* if (V, E) is a graph and H is a set of variables of (V, E) . For a set or a list S , $|S|$ denotes the number of elements in S . Let p be a graph pattern. Also, $V(p)$, $E(p)$, and $H(p)$ denote the sets of all vertices, edges, and variables of p , respectively. Moreover, $\lambda_p(u)$, $\delta_p(e)$, and $x_p(h)$ denote the vertex label of $u \in V(p)$, the edge label of $e \in E(p)$, and the variable label of $h \in H(p)$, respectively.

A graph pattern p' is said to be a *subgraph pattern* of p if $V(p') \subseteq V(p)$, $E(p') \subseteq E(p)$, and $H(p') \subseteq H(p)$. Below we regard a standard graph as a graph pattern with no variable. Let p and q be graph patterns. We say that p is *isomorphic* to q if there is a bijection $\psi : V(p) \rightarrow V(q)$ such that (1) for any $v \in V(p)$, $\lambda_p(v) = \lambda_q(\psi(v))$, (2) $\{u, v\} \in E(p)$ if and only if $\{\psi(u), \psi(v)\} \in E(q)$, (3) for any $\{u, v\} \in E(p)$, $\delta_p(\{u, v\}) = \delta_q(\{\psi(u), \psi(v)\})$, and (4) for $\ell \geq 1$, $(v_1, v_2, \dots, v_\ell) \in H(p)$ if and only if $(\psi(v_1), \psi(v_2), \dots, \psi(v_\ell)) \in H(q)$. We say that p is *subgraph isomorphic* to q if p is isomorphic to a subgraph pattern of q .

In an outerplanar embedding of an outerplanar graph G , an edge of G is *external* if it borders the outer face.

Definition 1 A graph pattern p is said to be an *externally extensible outerplanar graph pattern* (*eeo-graph pattern* for short) if p satisfies the following conditions: (1) Every variable has exactly 2 vertices, (2) $E(p) \cap E_H(p) = \emptyset$, where $E_H(p) = \{\{u, v\} \mid (u, v) \in H(p)\}$, and (3) the graph $G_p = (V(p), E(p) \cup E_H(p))$ of p is a connected outerplanar graph, and all edges in $E_H(p)$ are external.

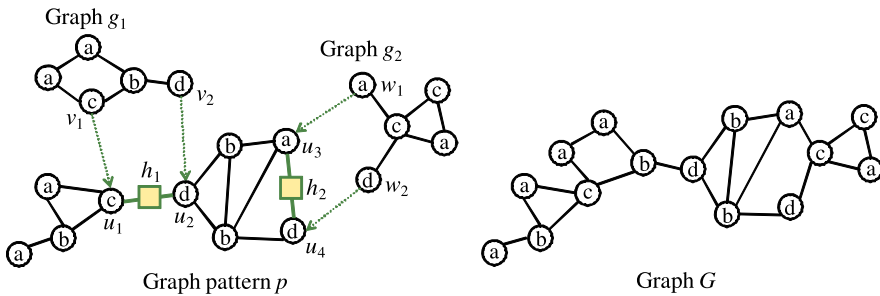


Fig. 2 A graph pattern p and graphs g_1, g_2, G over a vertex label set $\{a, b, c, d\}$

The variables of an eeo-graph pattern p are called *external variables*. Moreover, an external variable $h = (u, v)$ in p is called a *bridge variable* if $\{u, v\}$ is a bridge in G_p , and a *block variable* otherwise. If G_p is biconnected, p is called a biconnected eeo-graph pattern.

Definition 2 Let x be a variable label in X . Let p and q be eeo-graph patterns and $h = (v_1, v_2)$ a variable of p labelled with x . Let $\sigma = (u_1, u_2)$ be a list of two distinct vertices in q . The form $x := [q, \sigma]$ is called an eeo-binding for x if (1) $\lambda_q(u_1) = \lambda_p(v_1)$ and $\lambda_q(u_2) = \lambda_p(v_2)$, and (2) if h is a block variable of p , all edges on a path between u_1 and u_2 are bridges.

A new graph pattern $p\{x := [q, \sigma]\}$ is obtained by applying the eeo-binding $x := [q, \sigma]$ to p in the following way. Let $h = (v_1, v_2)$ be a variable in p with the variable label x . For $h = (v_1, v_2)$, we attach q to p by removing h from $H(p)$ and by identifying v_1 and v_2 with u_1 and u_2 of q , respectively. The new graph pattern $p\{x := [q, \sigma]\}$ is also an eeo-graph pattern.

Let p, q_1, \dots, q_m be eeo-graph patterns. An *eeo-substitution* for p is a finite collection of eeo-bindings $\{x_1 := [q_1, \sigma_1], \dots, x_m := [q_m, \sigma_m]\}$ where x_1, \dots, x_m are mutually distinct variable labels in X and each g_i ($1 \leq i \leq m$) has no variable labelled with a variable label in $\{x_1, \dots, x_m\}$. For an eeo-graph pattern p and an eeo-substitution θ for p , $p\theta$ denotes the eeo-graph pattern obtained from p and θ by applying all the eeo-bindings in θ to p simultaneously.

We give an example of eeo-substitutions. In Fig. 2, we give a graph pattern p that has variables $h_1 = (u_1, u_2)$ and $h_2 = (u_3, u_4)$, so that the graph $p\{x_p(h_1) := [g_1, (v_1, v_2)], x_p(h_2) := [g_2, (w_1, w_2)]\}$ is isomorphic to G .

Given a graph pattern p and a graph G , an operator that decides whether or not p matches G is called a *matching operator*. An isomorphism (resp. a subgraph isomorphism) matching operator returns a “true” if there is a substitution θ such that $p\theta$ is isomorphic (resp. subgraph isomorphic) to G , and “false” otherwise. Because the subgraph isomorphism problem for outerplanar graphs is NP-complete, the problem of deciding whether a subgraph isomorphism matching operator returns a “true” or “false” is hard to solve. On the other hand, the subgraph isomorphism problem for biconnected outerplanar graphs is solvable in cubic time [5]. In this

paper, we use a restricted subgraph isomorphism introduced by [4] as a matching operator.

Definition 3 For two graphs G and H , we say that G is bridge and block preserving (BBP) subgraph isomorphic to H if there is a subgraph isomorphism ψ from G to H that maps (i) the set of bridges of G to the set of bridges of H and (ii) different blocks of G to different blocks of H .

For an eeo-graph pattern p and an outerplanar graph G , we say that p matches G if there is an eeo-substitution θ such that $p\theta$ is BBP subgraph isomorphic to G .

For example, in Fig. 1, from p and $\theta = \{x_p(h_1) := [g_1, (v_1, v_2)], x_p(h_2) := [g_2, (w_1, w_2)], x_p(h_3) := [g_3, (z_1, z_2)]\}$, we obtain the outerplanar graph $p\theta$ that is isomorphic to G . The graph G is BBP subgraph isomorphic to H such that blocks B_1, B_2, B_3 , and B_4 of G correspond to B_5, B_6, B_7 , and B_8 of H , respectively. Therefore, p matches both G and H .

3 A Matching Algorithm for EEO-Graph Patterns

First of all, we give an outline of a polynomial time algorithm for solving the matching problem for a given eeo-graph pattern p and a given outerplanar graph G . The idea of our algorithm for the matching problem is similar to the polynomial time matching algorithm for block preserving graph patterns in the works of Sasaki et al. [7] and Yamasaki et al. [10]. First we transform p and G into labelled tree structure representations $t(p)$ and $T(G)$, respectively, and fix one vertex of $t(p)$ as its root. Next, for every vertex of $T(G)$, we specify it as the root of $T(G)$, and proceed to construct correspondences between all vertices $T(G)$ and $t(p)$ in the bottom up manner, that is, from the leaves to the root of $T(G)$. In the process of the algorithm, we have to decide whether or not a block of G is matched by a subgraph pattern of p that has block variables.

An eeo-graph pattern p with a distinguished vertex $r \in V(p)$ is called a rooted eeo-graph pattern and denoted by p^r . Similarly, an outerplanar graph G with a distinguished vertex $s \in V(G)$ is called a rooted outerplanar graph and denoted by G^s . We say that a rooted eeo-graph pattern p^r matches a rooted outerplanar graph G^s if p matches G , so that r is mapped to s . $\mathcal{EOP}_{\Lambda, \Delta}$ (resp. $\mathcal{O}_{\Lambda, \Delta}$) denotes the set of all eeo-graph patterns (resp. the set of all connected outerplanar graphs) over a vertex label set Λ and an edge label set Δ . Moreover, $\mathcal{EOP}_{\Lambda, \Delta}^{rb}$ (resp. $\mathcal{O}_{\Lambda, \Delta}^{rb}$) denotes the set of all rooted biconnected eeo-graph patterns (resp. the set of all rooted biconnected outerplanar graphs) over a vertex label set Λ and an edge label set Δ .

We give a polynomial time algorithm for solving the following problem.

Matching Problem for Rooted Biconnected EEO-Graph Patterns

Input: A rooted biconnected eeo-graph pattern $p^r \in \mathcal{EOP}_{\Lambda, \Delta}^{rb}$ and a rooted biconnected outerplanar graph $G^s \in \mathcal{O}_{\Lambda, \Delta}^{rb}$.

Problem: Decide whether or not p^r matches G^s .

For $p^r \in \mathcal{EOP}_{\Lambda, \Delta}^{rb}$ and $G^s \in \mathcal{O}_{\Lambda, \Delta}^{rb}$, let $n = |V(p^r)|$ and $N = |V(G^s)|$. Hereafter, we assume that $V(p^r) = \{1, 2, \dots, n\}$ with $r = 1$ and $V(G^s) = \{1, 2, \dots, N\}$ with $s = 1$, and that sequences $(1, 2, \dots, n)$ and $(1, 2, \dots, N)$ are clockwise or counter-clockwise orders of boundaries of outerplanar embeddings of p^r and G^s .

Lemma 4 For $p^r \in \mathcal{EOP}_{\Lambda, \Delta}^{rb}$ and $G^s \in \mathcal{O}_{\Lambda, \Delta}^{rb}$, if p^r matches G^s , there is an eeo-substitution θ for p^r , and a subgraph isomorphism $\psi : V(p\theta) \rightarrow V(G)$ such that $\psi(1) = 1$ and the vertex sequence $\psi(2), \psi(3), \dots, \psi(n)$ is either an increasing or decreasing number sequence.

Proof Let $C = (i_1, i_2, \dots, i_m)$ be a simple cycle of G^s such that $i_1 = 1$. We assume that there is an index k such that $2 < k < m$, i_2, \dots, i_k is an increasing number sequence, and $i_{k+1} < i_k$. Then, there is an index h such that $1 \leq h < k$ and $i_h < i_{k+1} < i_{h+1}$. On the outer cycle of G^s , there is a path reaching from i_h to i_{k+1} , denoted by P_1 , and a path reaching from i_{k+1} to i_{h+1} , denoted by P_2 . Let P_3 be the path $(i_{k+1}, \dots, i_m, i_1)$ on C , and P_4 the path $(i_k, i_{k+1}, \dots, N, 1)$ on the outer cycle of G^s . Since the outer cycle of G^s and C are simple cycles, P_3 does not contain i_k and P_4 does not contain i_{k+1} . Since P_3 and P_4 share at least one vertex, let ℓ be the first cross point of P_3 and P_4 . Let P_5 be a path consisting of ℓ, \dots, N on the outer cycle of G^s and i_1, \dots, i_{h+1} on C .

Suppose that $k \neq h + 1$. We note that P_5 does not contain i_k and i_{k+1} . Let P_6 be the path $(i_{h+1}, i_{h+2}, \dots, i_k)$ on C . Then a subgraph of G^s , which consists of P_2, P_3, P_4, P_5, P_6 and the edge $\{i_k, i_{k+1}\}$, is homeomorphic to K_4 (the complete graph on four vertices). No outerplanar graph contains any subgraph homeomorphic to K_4 . Suppose that $k = h + 1$. We note that $i_h > i_1$ and $i_h \neq \ell$. Then, a subgraph of G^s consisting of P_1, P_3, P_4, P_5 , and the edge $\{i_k, i_{k+1}\}$ is homeomorphic to K_4 . This contradicts the fact that G^s is an outerplanar graph. Therefore, i_2, \dots, i_m is either an increasing or decreasing number sequence.

From this fact, for any biconnected outerplanar subgraph g of G^s containing s , the outer cycle of g has the vertex sequence as an increasing number sequence. From the definition of eeo-graph patterns, since the order of vertices of the outer cycle of p^r is not changed by any eeo-substitution, consequently, the lemma holds. \square

For vertices i and i' ($i < i'$) in p^r , $p^r[i, i']$ denotes the subgraph pattern obtained from the induced subgraph $p[\{i, i + 1, \dots, i'\}]$ by removing a variable $(i, i') \in H(p)$ if it exists. For example, in Fig. 3 we give a rooted biconnected eeo-graph pattern p^r and its subgraph patterns $p^r[1, 5]$ and $p^r[5, 8]$. For vertices i and i' ($i < i'$) in p^r , and vertices j and j' ($j < j'$) in G^s , we say that $p^r[i, i']$ matches $G^s[j, j']$ if there is an eeo-substitution θ for $p^r[i, i']$ and a subgraph isomorphism $\psi : V(p^r[i, i']\theta) \rightarrow$

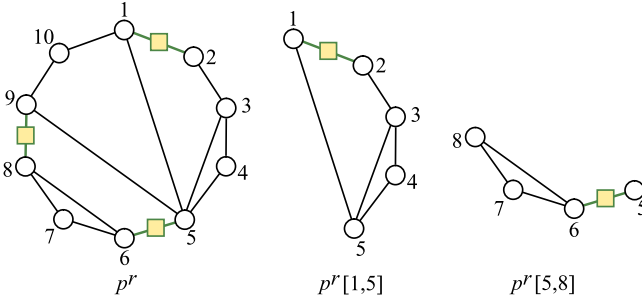


Fig. 3 A rooted biconnected eeo-graph pattern and its subgraph patterns

$V(G^s[j, j'])$ such that $\psi(i) = j$ and $\psi(i') = j'$. For vertices i and i' ($i < i'$) in p^r , the *correspondence-set* (*C-set* for short) of the pair (i, i') , denoted by $CS(i, i')$, is the set of all pairs (j, j') of vertices in G^s such that $p^r[i, i']$ matches $G^s[j, j']$.

Lemma 5 For a vertex i in p^r where $1 \leq i < n$, and vertices j and j' ($j < j'$) in G^s , $(j, j') \in CS(i, i + 1)$ if and only if either of the following conditions holds.

1. $\{i, i + 1\} \in E(p^r)$, $\{j, j'\} \in E(G^s)$, $\lambda_p(i) = \lambda_G(j)$, $\lambda_p(i + 1) = \lambda_G(j')$, and $\delta_p(\{i, i + 1\}) = \delta_G(\{j, j'\})$.
2. $(i, i + 1) \in H(p^r)$, $\lambda_p(i) = \lambda_G(j)$, and $\lambda_p(i + 1) = \lambda_G(j')$.

Proof If $\{i, i + 1\} \in E(p^r)$, it is easy to see that statement (1) holds. If $(i, i + 1) \in H(p^r)$, $p^r[i, i + 1]$ consists of two vertices, i and $i + 1$, and has no edge and no variable. Therefore, statement (2) holds. □

Statement (2) of the above lemma means a variable $(i, i + 1)$ supplements any subgraph $G^s[j, j']$ such that $\lambda_p(i) = \lambda_G(j)$ and $\lambda_p(i + 1) = \lambda_G(j')$. We note that any variable in p^r forms either $(i, i + 1)$ ($1 \leq i < n$) or $(1, n)$.

Let i and i' be vertices in p^r such that $i' - i > 1$ and they border on the same inner face; in other words, if an edge $\{i, i'\}$ exists, $\{i, i'\}$ maintains the outerplanarity of p^r . The *chordless path* of $p^r[i, i']$, denoted by $P(p^r[i, i'])$, is a simple path (i_1, i_2, \dots, i_m) such that i_1, i_2, \dots, i_m is an increasing number sequence satisfying that $m > 2$, $i_1 = i$, $i_m = i'$, and, except for edges and variables constituting the path, there is possibly only one edge $\{i_1, i_m\}$ or variable (i_1, i_m) between all vertices in the path.

For example, in Fig. 3, we can see that chordless paths of $p^r[1, 5]$ and $p^r[5, 8]$ are $(1, 2, 3, 5)$ and $(5, 6, 8)$, respectively. It is easy to show the following lemma.

Lemma 6 Let i and i' be vertices in p^r such that $i' - i > 1$ and they border on a same inner face, and let $P(p^r[i, i']) = (i_1, i_2, \dots, i_m)$. If $\{i, i'\} \notin E(p^r)$, then the vertices i_1, i_2, \dots, i_m are cutpoints of $p^r[i, i']$.

Lemma 7 *Let i and i' be vertices in p^r such that $i' - i > 1$ and they border on a same inner face, and j and j' vertices in G^s such that $j < j'$. Let $P(p^r[i, i']) = (i_1, i_2, \dots, i_m)$. Then, $(j, j') \in CS(i, i')$ if and only if the following conditions hold.*

1. *If $\{i, i'\} \in E(p^r)$, $\{j, j'\} \in E(G^s)$ and $\delta_p(\{i, i'\}) = \delta_G(\{j, j'\})$.*
2. *There is a vertex j'' in G^s such that $j < j'' < j'$, $(j, j'') \in CS(i_1, i_{m-1})$ and $(j'', j') \in CS(i_{m-1}, i_m)$.*

Proof Suppose that the if-statement holds. There is an eeo-substitution θ_1 for $p^r[i_1, i_{m-1}]$ (resp. θ_2 for $p^r[i_{m-1}, i_m]$), and a subgraph isomorphism ψ_1 from g_1 to $G^s[j, j'']$ where $g_1 = p^r[i_1, i_{m-1}]\theta_1$ (resp. ψ_2 from g_2 to $G^s[j'', j']$ where $g_2 = p^r[i_{m-1}, i_m]\theta_2$) such that $\psi_1(i_1) = j$ and $\psi_1(i_{m-1}) = j''$ (resp. $\psi_2(i_{m-1}) = j''$ and $\psi_2(i_m) = j'$). Let θ be the union of θ_1 and θ_2 , and $g = p^r[i, i']\theta$. Let ψ be the injection from $V(g)$ to $V(G^s[j, j'])$ such that $\psi(k) = \psi_1(k)$ for each $k \in V(g_1)$ and $\psi(k) = \psi_2(k)$ for each $k \in V(g_2)$. Because statement (1) holds and, from Lemma 6, there is possibly only one edge (i, i') between $V(g_1) - \{i_{m-1}\}$ and $V(g_2) - \{i_{m-1}\}$, we see that for all vertices k and k' in g , if $\{k, k'\} \in E(g)$, then $\{\psi(k), \psi(k')\} \in E(G^s)$ where $\delta_g(\{k, k'\}) = \delta_G(\{\psi(k), \psi(k')\})$. Therefore, ψ is a subgraph isomorphism from g to $G^s[j, j']$.

Conversely, suppose that there is an eeo-substitution θ for $p^r[i, i']$, and a subgraph isomorphism ψ from g to $G^s[j, j']$ where $g = p^r[i, i']\theta$, such that $\psi(i) = j$ and $\psi(i') = j'$. It is easy to see that statement (1) holds. By assuming $\{i, i'\} \in E(p^r)$, we regard g and $G^s[j, j']$ as biconnected outerplanar graphs. We suppose that all vertices on the outer cycle of g are identified as $1, 2, \dots, \ell$ such that i and i' are identified as 1 and ℓ , respectively, and i_{m-1} is identified as ℓ' in g . From Lemma 4, the vertex sequence $(\psi(1), \psi(2), \dots, \psi(\ell))$ is an increasing number sequence. Let $j'' = \psi(\ell')$, and $g[1, \ell']$ and $g[\ell', \ell]$ will be subgraph isomorphic to $G^s[j, j'']$ and $G^s[j'', j']$, respectively. Let θ_1 and θ_2 be subsets of θ , which consist of all eeo-bindings for variables in $p^r[i_1, i_{m-1}]$ and $p^r[i_{m-1}, i_m]$, respectively. Then, $g[1, \ell']$ (resp. $g[\ell', \ell]$) is obtained from $p^r[i_1, i_{m-1}]$ and θ_1 (resp. $p^r[i_{m-1}, i_m]$ and θ_2). Accordingly, $p^r[i_1, i_{m-1}]$ matches $G^s[j, j'']$ and $p^r[i_{m-1}, i_m]$ matches $G^s[j'', j']$. □

Lemma 8 *Let $\{i, i'\} \in E(p^r)$ be a diagonal in p^r such that $i < i'$, and j and j' vertices in G^s such that $j < j'$. Let $P(p^r[i, i']) = (i_1, i_2, \dots, i_m)$. Then, $(j, j') \in CS(i, i')$ if and only if $\{j, j'\} \in E(G^s)$ where $\delta_p(\{i, i'\}) = \delta_G(\{j, j'\})$ and there is an increasing number sequence (j_1, j_2, \dots, j_m) such that $j_1 = j$, $j_m = j'$, and $(j_k, j_{k+1}) \in CS(i_k, i_{k+1})$ for each k ($1 \leq k < m$).*

Proof It is immediate from Lemma 7. □

Lemma 9 *For $p^r \in \mathcal{EOP}_{\Lambda, \Delta}^r$ and $G^s \in \mathcal{O}_{\Lambda, \Delta}^s$, p^r matches G^s and the vertex sequence $(1, 2, \dots, n)$ is mapped to an increasing number sequence by a subgraph isomorphism $\psi : V(p^r\theta) \rightarrow V(G^s)$ for an eeo-substitution θ if and only if $(1, i) \in CS(1, n)$ for a vertex $i \in V(G^s)$.*

Algorithm: MATCHBLOCK- $\mathcal{EO}\mathcal{P}^{rb}_{\Lambda, \Delta}$;

Input: $p^r \in \mathcal{EO}\mathcal{P}^{rb}_{\Lambda, \Delta}$ and $G^s \in \mathcal{O}^{rb}_{\Lambda, \Delta}$;

Output: TRUE or FALSE;

begin

- 1: Let $n = |V(p^r)|$ and $N = |V(G^s)|$;
- 2: Let $(1, \dots, n)$ (resp. $(1, \dots, N)$) be the clockwise order of boundary of p^r (resp. G^s);
We assume that $r = 1$ and $s = 1$.
- 3: **foreach** (i, i') with $1 \leq i < i' \leq n$ s.t. $\{i, i'\} \in E(p^r)$ or $(i, i') \in H(p^r)$ **do begin**
- 4: $CS_d(i, i') := \emptyset$;
- 5: **if** $i' = i + 1$ and $\{i, i'\} \in E(p^r)$ **then**
- 6: **foreach** $\{j, j'\} \in E(G^s)$ s.t. $j < j', i \leq j \leq i + N - n, i' \leq j' \leq j' + N - n$,
 $\lambda_p(i) = \lambda_G(j)$, $\lambda_p(i') = \lambda_G(j')$ and $\delta_p(\{i, i'\}) = \delta_G(\{j, j'\})$ **do**
- 7: add (j, j') to $CS_d(i, i')$
- 8: **else if** $i' = i + 1$ and $(i, i') \in H(p^r)$ **then**
- 9: **foreach** (j, j') with $j < j', i \leq j \leq i + N - n$,
 $i' \leq j' \leq j' + N - n, \lambda_p(i) = \lambda_G(j)$, and $\lambda_p(i') = \lambda_G(j')$ **do**
- 10: add (j, j') to $CS_d(i, i')$
- 11: **end**;
- 12: Let S is an empty stack;
- 13: **for** $i' := n - 1$ **downto** 3 **do if** $\{1, i'\} \in E(p^r)$ **then** push($(1, i')$, S);
- 14: **for** $i := 2$ **to** $n - 1$ **do**
- 15: **for** $i' := n$ **downto** $i + 2$ **do if** $\{i, i'\} \in E(p^r)$ **then** push((i, i') , S);
- 16: **while** S is not empty **do begin**
- 17: $(i, i') := \text{pop}(S)$;
- 18: **foreach** $\{j, j'\} \in E(p^r)$ s.t. $i \leq j \leq i + N - n$ and
 $j' = \max\{k \mid j < k, i' \leq k \leq i' + N - n \text{ and } \{j, k\} \in E(G^s)\}$ **do begin**
- 19: $C := \text{MATCHSUBBLOCK}(p^r[i, i'], G^s[j, j'])$;
- 20: **foreach** $k \in C$ **do** add (j, k) to $CS_d(i, i')$
- 21: **end**
- 22: **end**;
- 23: $C := \text{MATCHSUBBLOCK}(p^r[1, n], G^s[1, N])$;
- 24: **if** $C \neq \emptyset$ **then return** TRUE;
- 25: Let $(1, \dots, n)$ be the counterclockwise order of boundary of p^r ;
- 26: Do lines 3–24;
- 27: **return** FALSE

end.

Fig. 4 A pattern matching algorithm for biconnected eeo-graph patterns

Proof If $\{1, n\} \in E(p^r)$, it is easy to see that the lemma holds. If $(1, n) \in H(p^r)$, the variable $(1, n)$ can supplement the path $(i, i + 1, \dots, N, 1)$, therefore, p^r matches G^s . \square

Using Lemmas 4–9, we give an algorithm for solving the matching problem for biconnected eeo-graph patterns. For any vertex i in p^r , any vertex j in G^s corresponding to i satisfies $i \leq j \leq i + N - n$. Therefore, to decide whether or not p^r matches G^s , we only need to compute subsets of C-set, $CS_d(i, i') = \{(j, j') \in CS(i, i') \mid i \leq j \leq i + N - n \text{ and } i' \leq j' \leq i' + N - n\}$ for each $\{i, i'\} \in E(p^r)$

Procedure: MATCHSUBBLOCK;

Input: an eeo-subgraph pattern $p^r[i, i']$ and an outerplanar subgraph $G^s[j, j']$;

Output: the set $\{k \in V(G^s) \mid j < k \leq j' \text{ and } p^r[i, i'] \text{ matches } G^s[j, k]\}$;

begin

1: $P(p^r[i, i']) := (i_1, i_2, \dots, i_m)$; // $i_1 = i, i_m = i'$.

2: $CS_{i,j}(i_1) := \{j\}$;

3: **for** $\ell := 2$ **to** m **do begin**

4: $CS_{i,j}(i_\ell) := \emptyset$; // $CS_{i,j}(i_\ell)$ means the set $\{k \in V(G^s) \mid (j, k) \in CS_d(i_1, i_\ell)\}$.

5: **foreach** $k \in CS_{i,j}(i_{\ell-1})$ **do**

6: **foreach** $(k, k') \in CS_d(i_{\ell-1}, i_\ell)$ **do**

7: **if** $\ell < m$ **then** add k' to $CS_{i,j}(i_\ell)$

8: **else if** $\{i_1, i_m\} \in E(p^r)$ and $\{j, k'\} \in E(G^s)$ s.t. $\delta_p(\{i_1, i_m\}) = \delta_G(\{j, k'\})$ **then**

9: add k' to $CS_{i,j}(i_m)$

10: **else if** $(i_1, i_m) \in H(p^r)$ **then** add k' to $CS_{i,j}(i_m)$

11: **end;**

12: **return** $CS_{i,j}(i')$

end.

Fig. 5 A procedure for computing a subset of $CS_d(i, i')$ with respect to $G^s[j, j']$

and $(i, i') \in H(p^r)$. Hence, we give a pattern matching algorithm $\text{MATCHBLOCK-}\mathcal{EOP}_{\Lambda, \Delta}^{rb}$ in Fig. 4 for computing the sets $CS_d(i, i')$ for all edges $\{i, i'\} \in E(p^r)$ and variables $(i, i') \in H(p^r)$ by using a dynamic programming manner. The algorithm assigns all C-sets of pair $(i, i+1)$ where $1 \leq i < n$ first, and it assigns each C-set of pair (i, i') such that $\{i, i'\} \in E(p^r)$ is a diagonal and for its chordless path (i_1, i_2, \dots, i_m) , the C-sets of all pairs (i_k, i_{k+1}) have been already assigned for all k ($1 \leq k < m$). For an eeo-subgraph pattern $p^r[i, i']$ and an outerplanar subgraph $G^s[j, j']$, Procedure MATCHSUBBLOCK in Fig. 5 computes a subset of $CS_d(i, i')$, the set $\{(j, k) \in CS_d(i, i') \mid k \leq j'\}$. The assignment of C-sets terminates when the C-set of pair $(1, n)$ is assigned, and if $(1, i) \notin CS_d(1, n)$ for all $i \in V(G^s)$, we identify vertices of p^r in reverse and assign C-sets again.

Then, we have the following lemma.

Lemma 10 *For a rooted biconnected eeo-graph pattern p^r and a rooted biconnected outerplanar graph G^s , the problem of deciding whether or not p^r matches G^s can be correctly solved in $O(nN^2)$ time, where $n = |V(p)|$ and $N = |V(G)|$.*

Proof Since $|E(p)| + |H(p)| \leq 2|V(p)| - 3$, we need $O(n(N-n)^2)$ time at the initial stage (lines 3–11) of $\text{MATCHBLOCK-}\mathcal{EOP}_{\Lambda, \Delta}^{rb}$. Let $p^r[i, i']$ be a subgraph pattern of p^r and $P(p^r[i, i']) = (i_1, i_2, \dots, i_m)$. In Procedure MATCHSUBBLOCK, each vertex i_ℓ ($1 \leq \ell \leq m$) is assigned at most $N - n$ vertices of G^s , and therefore, MATCHSUBBLOCK works in $O(m(N-n))$ time. Since the total lengths of chordless paths of diagonals in p^r is $|E(p)| + |H(p)| - 1$, we need $O(nN(N-n))$ time at lines 16–22 of $\text{MATCHBLOCK-}\mathcal{EOP}_{\Lambda, \Delta}^{rb}$. Therefore, the total time is $O(nN^2)$ time. \square

Theorem 11 *The matching problem for an eeo-graph pattern $p \in \mathcal{EOP}_{\Lambda, \Delta}$ and an outerplanar graph $G \in \mathcal{O}_{\Lambda, \Delta}$ is computable in $O(nN^3)$ time, where n and N are the numbers of vertices of p and G , respectively.*

Proof Let u and v be vertices of rooted eeo-graph pattern p^r and rooted outerplanar graph $G^s[v]$, respectively. From Lemma 10 and the proof of Corollary 1 in the work of Yamasaki et al. [10], to decide whether or not $p^r[u]$ matches $G^s[v]$, we need $O(c_u c_v^2)$ time where c_u and c_v are the numbers of children of u and v , respectively. Then, the runtime for deciding whether or not p^r matches G^s is $O(nN^2)$ time, and consequently, the total time is $O(nN^3)$ time. \square

4 Frequent Pattern Mining on Chemical Datasets

To show the expressiveness of our graph pattern class $\mathcal{EOP}_{\Lambda, \Delta}$, we have implemented our matching algorithm with a frequent pattern mining method of Yamasaki et al. [10], and experimented on subsets of the DTP AIDS antiviral screen database [2], which contains 42,689 chemical compounds.

Let S be a finite subset of $\mathcal{O}_{\Lambda, \Delta}$ and p a eeo-graph pattern in $\mathcal{EOP}_{\Lambda, \Delta}$. Then, $O_S(p)$ denotes the set of outerplanar graphs in S that are matched by p . The *frequency* of p with respect to S , denoted by $supp_S(p)$, is defined as $supp_S(p) = |O_S(p)|/|S|$. Let t be a real number where $0 < t \leq 1$. A eeo-graph pattern $p \in \mathcal{EOP}_{\Lambda, \Delta}$ is *t -frequent* with respect to S if $supp_S(p) \geq t$. We call this real number t a *frequency threshold*.

The number of chemical compounds that can be expressed by connected outerplanar graphs is 35,515 (about 83.2%). The DTP AIDS Antiviral Screen program has checked tens of thousands of chemical compounds for evidence of anti-HIV activity. Compounds in the database are classified into three categories, confirmed active (CA), confirmed moderately active (CM), and confirmed inactive (CI). We used a set of 302 CA compounds, denoted by S_{CA} , all of which can be expressed by connected outerplanar graphs. We applied our method to the set S_{CA} with frequency threshold 0.3, and generated frequent bpo-graph patterns [7] and eeo-graph patterns. From discovered bpo-graph patterns (resp. eeo-graph patterns), we extracted graph patterns that are maximal with respect to frequency threshold t , i.e., t -frequent bpo-graph patterns (resp. eeo-graph patterns) p satisfying that there is no t -frequent bpo-graph pattern (resp. eeo-graph patterns) q such that $L(q) \subsetneq L(p)$ and $supp_{S_{CA}}(p) = supp_{S_{CA}}(q)$, where $L(p) = \{G \in \mathcal{O}_{\Lambda, \Delta} \mid p \text{ matches } G\}$.

We found 836 frequent bpo-graph patterns and 18,620 frequent eeo-graph patterns that are maximal with respect to frequencies. Moreover, from frequent eeo-graph patterns, we picked out eeo-graph patterns with no variable, which correspond to subgraphs obtained by applying a frequent outerplanar subgraph mining algorithm of Horvath et al. [4]. There are 28 of these subgraphs. For these three types of graph patterns, we computed frequencies with respect to a set of 34,369 CI compounds, denoted by S_{CI} , all of which can be expressed by connected outerplanar graphs. We extracted graph patterns the frequencies of which for S_{CI} are lower

	eoo-graph pattern	bpo-graph pattern	subgraph
1	0.472	0.322	0.321
2	0.468	0.322	0.319
3	0.466	0.321	0.317
4	0.464	0.317	0.316
5	0.463	0.317	0.314
6	0.462	0.312	0.311
7	0.450	0.309	0.300
8	0.450	0.308	–
9	0.449	0.308	–
10	0.449	0.302	–

Fig. 6 The top 10 scores for each type of graph pattern. For a graph pattern p , the score of p is $supp_{S_{CA}}(p) - supp_{S_{CI}}(p)$

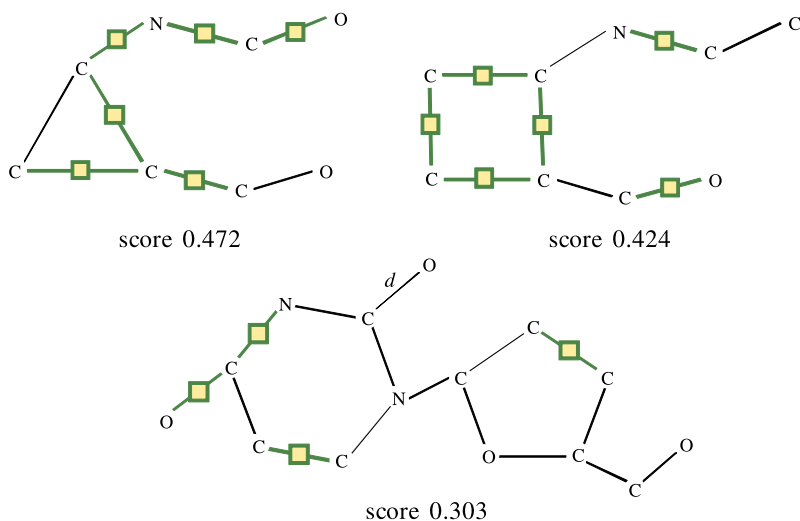


Fig. 7 Examples of eoo-graph patterns discovered in a dataset of the DTP AIDS antiviral screen

than 0.05. Accordingly, we obtained 15 bpo-graph patterns, 4,113 eoo-graph patterns, and 7 subgraphs. The maximum numbers of vertices of obtained bpo-graph patterns, eoo-graph patterns, and subgraphs are 9, 15, and 8, respectively.

For each obtained graph pattern p , we computed $supp_{S_{CA}}(p) - supp_{S_{CI}}(p)$, called the score of p . Figure 6 shows the top 10 scores for each type of graph pattern. The number of obtained bpo-graph patterns is larger than the number of subgraphs, but there is little difference between top scores of these two types of graph patterns. The number of obtained eoo-graph patterns is larger than the numbers of the other two types of graph patterns, and top scores of eoo-graph patterns are higher. The result shows that eoo-graph patterns can capture more detailed characteristics common to

active data than other outerplanar graph patterns. We show examples of discovered eeo-graph patterns in Fig. 7.

5 Conclusions

In this paper, we introduced externally extensible outerplanar graph patterns (eeo-graph patterns) and gave a polynomial time algorithm for deciding whether or not a given eeo-graph pattern matches a given connected outerplanar graph. Moreover, we have evaluated the expressiveness of the graph pattern class by experiments on chemical compound database. For future works aiming at efficient graph mining systems for real-world databases, we are studying graph pattern representations based on graph transformation systems.

References

1. Cook, D.J., Holder, L.: Mining Graph Data. Wiley-Interscience, New York (2007)
2. Developmental Therapeutics Program NCI/NIH: AIDS Antiviral Screen. <http://dtp.nci.nih.gov/>
3. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Mateo (2001)
4. Horváth, T., Roman, J., Wrobel, S.: Frequent subgraph mining in outerplanar graphs. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 197–206 (2006)
5. Lingas, A.: Subgraph isomorphism for biconnected outerplanar graphs in cubic time. Theoret. Comput. Sci. **63**, 295–302 (1989)
6. National Cancer Institute: Chemical dataset. <http://cactus.nci.nih.gov/>
7. Sasaki, Y., Yamasaki, H., Shoudai, T., Uchida, T.: Mining of frequent block preserving outerplanar graph structured patterns. In: Proceedings of the 17th International Conference on Inductive Logic Programming (ILP2007). Lect. Notes Artif. Intell., vol. 4894, pp. 239–253. Springer, Berlin (2008)
8. Suzuki, Y., Shoudai, T., Uchida, T., Miyahara, T.: Ordered term tree languages which are polynomial time inductively inferable from positive data. Theoret. Comput. Sci. **350**, 63–90 (2006)
9. Yamasaki, H., Yamada, T., Shoudai, T.: An expressive outerplanar graph pattern class and its efficient pattern matching algorithm. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 471–477 (2010)
10. Yamasaki, H., Sasaki, Y., Shoudai, T., Uchida, T., Suzuki, Y.: Learning block-preserving graph patterns and its application to data mining. Mach. Learning **76**(1), 137–173 (2009)

Setvectors – An Efficient Method to Predict Cache Contention

Michael Zwick

Abstract In this chapter, I present a new method called *Setvectors* to predict cache contention introduced by co-scheduled applications on a multicore processor system. Additionally, I propose a new metric to compare cache contention prediction methods. Applying this metric, I demonstrate that the *Setvector* method predicts cache contention with about the same accuracy as the most accurate state-of-the-art method. However, the Setvector method executes nearly 4000 times as fast. This chapter is a revised and extended version of Zwick et al. (Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, pp. 244–251, 2010), presented at the MultiConference of Engineers and Computer Scientists in Hong Kong.

Keywords Cache contention · Co-scheduling · Setvectors

1 Introduction

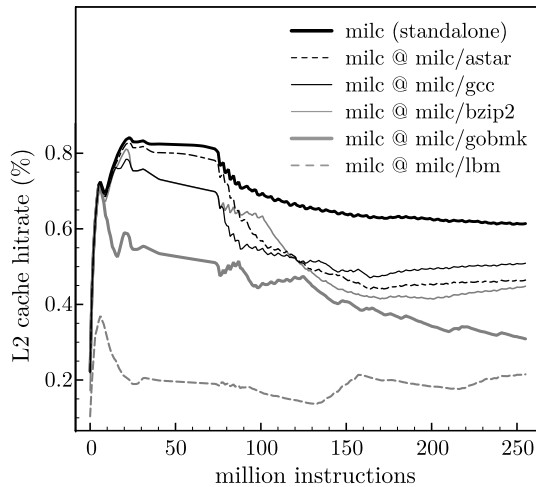
With multicore processors, chip manufacturers try to satisfy the ever increasing demand for computational power by parallelization on thread or process basis, making performance of computer systems more and more independent from the saturated processor clock speed. However, one important limitation that does not rely on processor clock speed, but on the computational power of the processor, is the ever increasing processor memory gap: Although both, processor and DRAM performance, grow exponentially over time, the performance difference between processor and DRAM grows exponentially, too. This happens due to the fact that “the exponent for processors is substantially larger than that for DRAMs” [7] and “the difference between diverging exponentials also grows exponentially” [7].

M. Zwick (✉)

Lehrstuhl für Datenverarbeitung, Technische Universität München, Arcisstr. 21, 80333 Munich, Germany

e-mail: zwick@tum.de

Fig. 1 L2 cache hitrate degradation for the *milc* SPEC2006 benchmark when co-scheduled with different applications



A way to deal with the exponentially diverging memory gap is to transform computational performance into memory hierarchy performance, making memory performance not only benefit from improvements of the memory hierarchy system, but also from better (and in a much higher rate evolving) processor technology. One possibility therefore is to spend computational power to find good application co-schedules that minimize overall cache contention. Reducing DRAM accesses by optimizing cache performance is a key issue in today's and tomorrow's computer architectures.

Fedorova [2] identified L2 cache performance as a most crucial factor regarding overall performance degradation in multicore processors. Figure 1 shows the effect of L2 cache contention on the SPEC2006 benchmark *milc*, running on one core of a dual core processor, while each of the applications *astar*, *gcc*, *bzip2*, *gobmk* and *lbm* is executed on the other core. It can easily be seen that the performance of *milc* heavily degrades when co-scheduled with the *lbm* benchmark, while other co-schedules have a much lower performance burden.

The most important requirement in order to optimize co-schedules for cache contention is a good method to *predict* cache contention of application co-schedules from specific application characteristics. Although a number of techniques have been investigated to predict L2 cache performance from some application characteristics for single core processors, only little effort has been spent so far to predict L2 cache performance of co-scheduled applications in a multicore scenario.

In this chapter, I propose a new method called *Setvectors* to predict cache contention in multicore processors. I compare this method to the *Activity vectors* proposed by Settle et al. [6] and the circular sequence based *Prob* model presented by Chandra et al. [1]. I show that the *Setvector* approach predicts optimal co-schedules with about the same accuracy as the best performing circular sequence based method does; however, it executes about 4000 times as fast. The remainder of this chapter is

organized as follows: Section 2 presents state-of-the-art techniques to predict cache contention; Section 3 introduces the *Setvector* method. In Sect. 4, I propose a new metric called *MRD* (mean ranking difference) to compare cache contention prediction techniques and discuss the parameters applied in the simulation. In Sect. 5, I present the results. Section 6 concludes this chapter.

2 State-of-the-Art Techniques to Predict Cache Contention

In this section, I present state-of-the-art techniques to predict cache contention in multiprocessor systems, namely *Activity vectors* introduced by Settle et al. [6] and Dhruva Chandra et al.’s *stack distance* based *FOA* (frequency of access) and *SDC* (stack distance competition) model [1] and the *circular sequence* based *Prob* (probability) model [1].

2.1 Settle et al.’s Activity Vectors

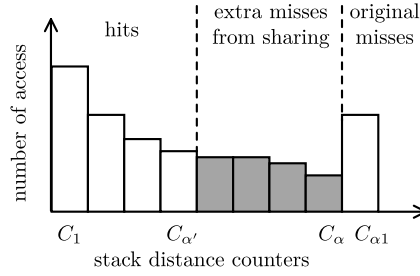
Alex Settle et al. studied processor cache activity and observed that “program behavior changes not only temporally, but also spatially with some regions hosting the majority of the overall cache activity” [6]. To exploit spatial behavior of cache activity to estimate cache contention, they divide the cache address space into groups of 32 so-called *super-sets* and count the number of accesses to each such super set. If, in a given time interval, the accesses to a super set exceed a predefined threshold, a corresponding bit in the so-called *Activity vector* is set to mark that super set as active.

To predict the optimal co-schedule *B*, *C* or *D* for a thread *A*, every bit in the Activity vector of *A* is logically AND-ed with the corresponding bit in each application *B*, *C* and *D*. The bits resulting from that operation are summed up for each thread combination $A \leftrightarrow B$, $A \leftrightarrow C$ and $A \leftrightarrow D$. As a co-schedule for *A*, Settle et al. select that thread in $\{B, C, D\}$ that yields the lowest value. Besides access count information, Settle et al. additionally apply *miss information* in the *Activity vectors*. To further distinguish Settle’s method from the circular sequence method however, I only regarded cache access information for my simulations; the presented results might look different, if miss rate information would be included in the Activity vectors.

2.2 Chandra et al.’s Stack Distance Based FOA and SDC Methods

Chandra et al. [1] propose to use *stack distances* to predict cache contention of co-scheduled tasks. Stack distances have originally been introduced by Mattson

Fig. 2 Stack distance histogram



et al. [5] to assist in the design of efficient storage hierarchies in virtual memory systems. Hill and Smith [3] showed that they can also be easily applied to evaluate cache memory systems.

The method assumes a cache with LRU (least recently used) replacement policy and works as follows: Given a cache with associativity α , the number of $\alpha + 1$ counters $C_1, \dots, C_{\alpha+1}$ have to be provided for each cache set to track the reuse behavior of cache lines. If, on a cache access, the requested cache line resides on position p of the LRU stack, counter C_p of the corresponding cache set is increased by one. If the cache access results in a miss, i.e. if the cache line has no corresponding entry on the LRU stack (and therefore the cache line does not reside in the cache), then counter $C_{\alpha+1}$ is increased. This procedure leads to a so-called *stack distance profile*, as it is depicted in Fig. 2. The stack distance profile characterizes the positions of cache lines on the LRU stack when accessing cache data.

Given a stack distance profile, the total number of accesses a to a specific cache set can simply be determined by summing up all C_i according to

$$a = \sum_{i=1}^{\alpha+1} C_i \quad (1)$$

and the cache miss rate P_m can be calculated by

$$P_m = \frac{C_{\alpha+1}}{\sum_{i=1}^{\alpha+1} C_i}. \quad (2)$$

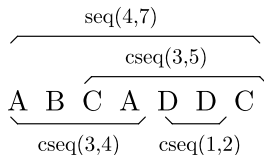
For caches with lower associativity α' , the miss rate can be computed by

$$P_m(\alpha') = \frac{\sum_{i=\alpha'}^{\alpha+1} C_i}{\sum_{i=1}^{\alpha+1} C_i}. \quad (3)$$

Chandra et al. exploit this equation to predict the cache miss rate under cache sharing. Let c be the size of a cache and let c'_x be the effective cache size that is available for thread x under cache sharing and let S be the number of cache sets, then Chandra et al. estimate the *effective associativity* α' of a task when sharing the cache with another task according to

$$\alpha' = \frac{c'_x}{S}. \quad (4)$$

Fig. 3 Relationship between sequences and circular sequences as presented similarly by Chandra et al. [1]. *A*, *B*, *C* and *D* depict different cache lines



For their *FOA* model, Chandra et al. calculate the effective cache size by

$$c'_x = \frac{\sum_{i=1}^{\alpha+1} C_{i,x}}{\sum_{y=1}^N \sum_{i=1}^{\alpha+1} C_{i,y}} \cdot c. \quad (5)$$

For the *SDC* model, Chandra et al. create a new stack distance profile by merging individual stack distance profiles to one profile and determine the effective cache space for each thread “proportionally to the number of stack distance counters that are included in the merged profile” [1]. The shaded region in Fig. 2 shows how the effective cache size is reduced by cache sharing. While the *FOA* and the *SDC* model both are heuristic models, Chandra et al. also developed an inductive probability model that is based on circular sequences (*cseq*) rather than on stack distances and is reported to deliver more accurate results than the *FOA* and *SDC* model.

2.3 Chandra et al.’s Circular Sequence Based Prob Method

Chandra et al.’s *Prob* method is based on circular sequences. Circular sequences are an extension to stack distances in that they do not only take into account the number of accesses to the different positions on the LRU stack, but also the number of cache accesses between accesses to equal positions on the LRU stack.

Therefore, Chandra et al. define a *sequence* $seq_x(d_x, n_x)$ as “a series of n_x cache accesses to d_x distinct line addresses by thread x , where all the accesses map to the same cache set” [1] and a *circular sequence* $cseq(d_x, n_x)$ as a sequence $seq_x(d_x, n_x)$ “where the first and the last accesses are to the same line and there are no other accesses to that address” [1]. Circular sequences can be regarded as stack distances that have each counter C augmented with an additional vector n to hold a histogram of accesses for each distance. Figure 3 illustrates the relationship between sequences and circular sequences when accessing cache lines *A*, *B*, *C* and *D*.

For their circular sequence based *Prob* model, Chandra et al. compute the number of cache misses m_x for a thread x when sharing the cache with an additional thread y by adding to the stand-alone cache misses $C_{\alpha+1}$ the values of the other counters $C_1 \dots C_\alpha$, each multiplied with the probability that the corresponding circular sequences $cseq(d_x, \bar{n}_x)$ will become a miss, where \bar{n}_x corresponds to the estimated *mean* n for a specific d :

$$m_x = C_{\alpha+1} + \sum_{d_x=1}^{\alpha} P_m(cseq_x(d_x, \bar{n}_x)) \times C_{d_x}. \quad (6)$$

Chandra et al. calculate the probability that the circular sequence $cseq(d_x, \bar{n}_x)$ will become a miss by summing up the probabilities that there are sequences $seq_y(d_y, E(n_y))$ in thread y with $\alpha - d_x + 1 \leq d_y \leq E(n_y)$, where $E(n_y)$ represents the expected value of n in the thread y :

$$P_m(cseq_x(d_x, \bar{n}_x)) = \sum_{d_y=\alpha-d_x+1}^{E(n_y)} P(seq_y(d_y, E(n_y))). \quad (7)$$

$E(n_y)$ is estimated by scaling \bar{n}_x proportionally to the ratio of accesses of y and x :

$$E(n_y) = \frac{\sum_{i=1}^{\alpha+1} C_{i_y}}{\sum_{i=1}^{\alpha+1} C_{i_x}} \cdot \bar{n}_x. \quad (8)$$

The probability of sequences $P(seq_y(d_y, E(n_y)))$, in short $P(seq(d, n))$, is calculated recursively according to

$$P(seq(\delta, \nu)) = \begin{cases} \sum_{i=1}^{\delta} P(cseq(i, *)) \cdot P(seq(\delta, \nu - 1)) \\ \quad + (1 - \sum_{i=1}^{\delta-1} P(cseq(i, *))) \\ \quad \times P(seq(\delta - 1, \nu - 1)) & \text{if } \nu > \delta > 1 \\ P(cseq(1, *)) \cdot P(seq(1, \nu - 1)) & \text{if } \nu > d = 1 \\ 1 - \sum_{i=1}^{\delta-1} P(cseq(i, *)) \\ \quad \times P(seq(\delta - 1, \delta - 1)) & \text{if } \nu = d > 1 \\ 1 & \text{if } \nu = d = 1. \end{cases} \quad (9)$$

The asterisk (*) in $cseq(i, *)$ denotes all possible values.

3 Setvector Based Cache Contention Prediction

In this section, I describe the Setvector method. First, I present the algorithm to obtain Setvectors. In a second step, I show how Setvectors can be used to predict cache contention.

3.1 Generating Setvectors

Setvectors are composed of cache set access frequencies \mathbf{a} and the number of different cache lines \mathbf{d} referenced within a specific amount of time, for example an operating system's timeslice. For this contribution, I collect one Setvector for every interval of 2^{20} instructions. According to the proposal in [9] where I presented Setvectors to predict L2 cache performance of stand-alone applications, I assume an L2 cache with 32 bit address length that uses b bits to code the *byte offset*, s bits to code the selection of the *cache set* and $k = 32 - s - b$ bits to code the *key* that has to be compared to the tags stored in the tag RAM. The Setvectors are gathered as follows. For every interval i of 2^{20} instructions do:

- First, set the 1×2^s sized vectors \mathbf{a} and \mathbf{d} to $\mathbf{0}$.
- Second, for every memory reference in the current interval, do:
 - Extract the set number σ from the address, e.g. by shifting the address k bits to the left and then unsigned-shifting the result $k + b$ bits to the right.
 - Extract the key from the address, e.g. by unsigned shifting the address $s + b$ bits to the right.
 - Increase $\mathbf{a}[\sigma]$.
 - In the list of the given set, determine whether the given key is already present.
 - If the key is already present, do nothing and proceed with the next address.
 - If the key is not in the list yet, add the key and increase $\mathbf{d}[\sigma]$.

This ends up with two 1×2^s dimensional vectors \mathbf{a} and \mathbf{d} . At index σ , \mathbf{a} holds the number of references to set σ and \mathbf{d} holds the number of memory references that map to set σ , but provide a different key.
- In a third step, subtract the cache associativity α from each element in \mathbf{d} and store the result in \mathbf{d}' . If the result gets negative, store 0 instead.
- In a fourth step, multiply each element of \mathbf{a} with the corresponding element in \mathbf{d}' and store the result in the 1×2^s dimensional Setvector \mathbf{s}_i .
- Finally, add \mathbf{s}_i as the i th column of matrix \mathbf{S} that holds in each column i the Setvector for interval i .
Process next interval.

3.2 Predicting Cache Contention with Setvectors

The compatibility of two threads for a time interval i can easily be predicted by just extracting \mathbf{s}_{i_x} from \mathbf{S}_x and \mathbf{s}_{i_y} from \mathbf{S}_y and calculating the dot product $\mathbf{s}_{i_x} \cdot \mathbf{s}_{i_y}$ of the Setvectors in order to obtain a single value. A low valued dotproduct implies a good match of the applications, a high dotproduct value suggests a bad match, i.e. a high level of cache interference resulting in many cache misses.

The dotproducts do not have any specific meaning like *number of additional cache misses*, as it is the case with Chandra's circular sequence based method. However, comparing the dotproducts of several thread combinations *in relation to each other* has been proven to be an effective way to predict which threads make a better match and which threads do not.

4 Evaluating Cache Contention Prediction Techniques – Simulation Setup

In order to prove the effectiveness of the Setvector method with its relative comparison of dotproducts, I compared it to Settle's Activity vector method and to Chandra's circular sequence based method. I refrained from additionally comparing the Setvector method to Chandra's stack distance based method, as Chandra already

Table 1 Parameterization of the MCCCsim simulator

Parameter	Private L1 cache	Shared L2 cache
Size	32 k	2 MB
Line size	128 Byte	128 Byte
Associativity	2 way	8 way
Hit time	1.0 ns	10.0 ns
Miss time	depends on L2	100.0 ns
Replacement	LRU	LRU

reported that the circular sequence based method outperformed the stack distance based methods – and the Setvector method showed nearly the same accuracy as the circular sequence based method.

To compare and evaluate cache contention prediction techniques, I generated tracefiles with memory accesses representing 512 million instructions for each of the ten SPEC2006 benchmark programs *astar*, *bzip2*, *gcc*, *gobmk*, *h264ref*, *hmmmer*, *lbm*, *mcf*, *milc* and *povray* applying the *Pin* toolkit [4]. Of these ten programs, I executed every 45 pairwise combinations on the MCCCsim multicore cache contention simulator [10] that had been parameterized as it is shown in Table 1.

For each program of each combination, I calculated the difference between the stand-alone memory access time and the memory access time when executed in co-schedule with the other application in units of picoseconds per instruction (*ps/instr.*). Figure 4 shows this additional penalty in the row *Simulated Penalty* for the *astar* application: If *astar* is co-scheduled with *hmmmer*, then a penalty of 2.3 *ps/instr.* is introduced to the *astar* application. When co-scheduled with *lbm*, *astar* suffers from a penalty of 134.7 *ps/instr.* As it can be seen from row *Simulated Penalty* in Fig. 4, I sorted the co-schedules according to the introduced penalty to yield a *ranking* such that a co-schedule with rank *i* is favorable to a co-schedule with rank *j > i*. If two co-schedules were equally valued, I sorted them by name to avoid nondeterministic rankings.

Similarly to the penalty ranking, I sorted the predictions I calculated for the *Setvector* method, Chandra et al.’s *circular sequence* method and Settle et al.’s *Activity vector* method. In Fig. 4, row *Prediction* shows an example of a prediction ranking.

To evaluate the prediction methods, I introduce an evaluation technique I call *mean ranking difference* (MRD): Figure 4 shows that I calculate the absolute difference between the rank determined by MCCCsim and the rank estimated by the prediction method for each combination. The results are summed up and divided by the total number of co-scheduled applications (9) to yield the average mean ranking distance (MRD), i.e. the mean number of ranks, a co-schedule’s prediction differs from the values obtained from MCCCsim.

I evaluated several variations of all three methods. Regarding Chandra et al.’s method, I was interested in comparing the predictions for the following variations:

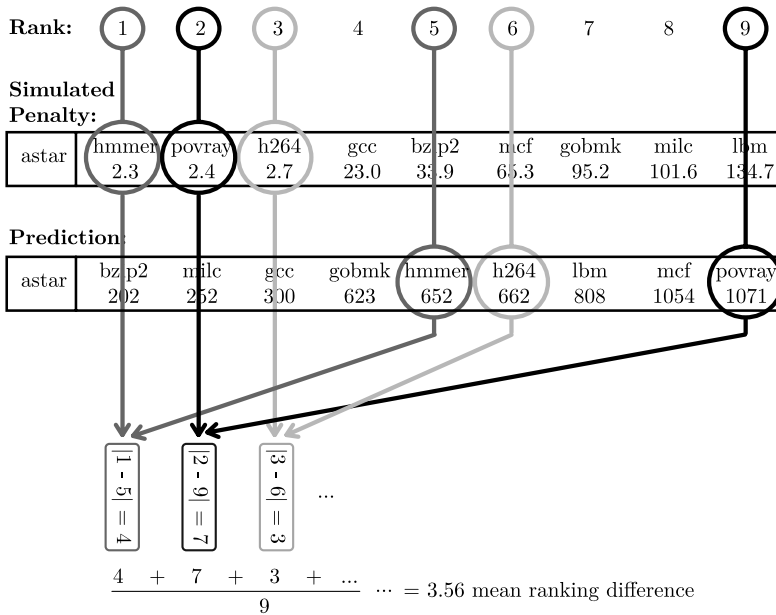


Fig. 4 Determination of the mean ranking difference (MRD) for *astar*

- *Chandra cseq, chunkset*: Prediction while applying only one circular sequence stack to a chunkset of instructions (i.e. interval of 2^{20} instructions).
- *Chandra cseq, Af(set)*: Prediction while applying a circular sequence stack to every cache set within an interval and measuring the memory access frequency on a *per cache set* basis.
- *Chandra cseq, Af(chunkset)*: Prediction while applying a circular sequence stack to every cache set within an interval without partitioning the memory access frequency on the cache sets, i.e. providing only *one* memory access frequency value per interval.

Settle et al. stated that “the low order bits of the cache set component of a memory address are used to index the activity counter associated with each cache super set” [6]. However, I expected that the method would achieve better results when using the *high* order bits to index the activity counters since addresses with equal high order bits are mapped to equal cache sets. Therefore, I evaluated the Activity vector method for these two variants naming them *high* respectively *low* (cf. Table 2).

Regarding the Setvector method, I was interested in analyzing the following variations (cf. Table 2):

- *diff. x access*: The Setvector method as presented in Sect. 3.
- *access*: Utilizing only the access frequency. This way, the performance of the Activity vector method can be estimated for the case that the number of supersets reaches its maximum (i.e. the over all number of sets) and the activity expresses

Table 2 Mean ranking difference (MRD) for each benchmark and method

Application	<i>Chandra cseq chunkset</i>	<i>Chandra cseq Af(set)</i>	<i>Chandra cseq Af(chunkset)</i>	<i>Activity low</i>	<i>Activity high</i>
astar	1.56	0.89	0.89	3.56	2.00
bzip	0.89	0.44	0.89	2.67	1.33
gcc	0.89	0.67	0.89	3.11	2.00
gobmk	0.67	0.67	0.44	3.11	3.33
h264ref	0.67	0.67	0.89	2.67	2.44
hmmer	0.89	0.67	1.11	2.89	2.44
lbm	1.11	0.67	1.33	4.00	2.22
mcf	0.44	0.22	1.33	3.11	2.22
milc	0.67	0.00	0.44	2.89	3.11
povray	2.00	0.89	0.89	2.67	2.67
average	0.98	0.58	0.91	3.07	2.38

Application	<i>Setvector diff. x access</i>	<i>Setvector access, add</i>	<i>Setvector access, mul</i>	<i>Setvector diff., add</i>	<i>Setvector diff., mul</i>
astar	0.67	0.67	0.44	0.89	0.89
bzip	0.67	0.67	0.22	0.44	0.89
gcc	0.89	0.89	0.67	0.67	0.67
gobmk	1.11	1.33	1.56	0.89	0.67
h264ref	0.44	0.44	0.44	1.11	1.11
hmmer	0.22	0.22	0.67	0.89	0.89
lbm	1.33	1.33	1.56	0.89	0.44
mcf	0.00	0.00	0.89	0.22	0.22
milc	0.22	0.44	0.89	0.22	0.44
povray	0.44	0.44	0.22	1.11	1.11
average	0.60	0.64	0.76	0.73	0.73

the number of accesses to a set and not just the one-bit information, whether or not a specific threshold has been reached.

- *diff*: Utilizing only the number of different cache lines that are mapped to the same cache set, i.e. ignoring any access frequency information.
- *add, mul*: Combining the vectors of two threads by applying either elementwise addition or multiplication and calculating the average of the elements afterwards, rather than by applying the dot product.

Table 3 Comparison of the execution times of the prediction methods

Method	Type	Nanoseconds per instruction for task										
		<i>astar</i>	<i>bzip2</i>	<i>gcc</i>	<i>gobmk</i>	<i>h264</i>	<i>hammer</i>	<i>lbn</i>	<i>mcf</i>	<i>milc</i>	<i>povery</i>	<i>average</i>
Chandra cseq,	setup	410.28	393.03	417.72	401.39	405.74	387.03	436.39	401.13	430.92	409.72	409.34
Af(set)	pred.	680.07	728.65	648.39	716.01	541.98	603.82	810.44	699.79	789.42	549.75	676.83
Chandra cseq,	setup	410.28	393.03	417.72	401.39	405.74	387.03	436.39	401.13	430.92	409.72	409.34
Af(chunkset)	pred.	679.09	724.70	649.47	848.25	736.89	595.34	802.37	708.18	980.43	735.81	746.05
Chandra cseq,	setup	410.28	393.03	417.72	401.39	405.74	387.03	436.39	401.13	430.92	409.72	409.34
chunkset	pred.	297.21	267.22	263.03	263.97	264.33	261.29	262.71	260.32	259.48	173.50	257.31
Activity vector,	setup	52.34	48.15	42.42	45.21	46.14	48.60	60.69	41.75	54.32	48.24	48.79
low	pred.	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Activity vector,	setup	52.34	48.15	42.42	45.21	46.14	48.60	60.69	41.75	54.32	48.24	48.79
high	pred.	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Setvector,	setup	55.51	52.69	44.05	50.76	48.18	51.69	83.08	46.79	61.35	51.32	54.54
diff. x access	pred.	0.16	0.17	0.16	0.17	0.16	0.16	0.18	0.17	0.17	0.17	0.17
Setvector,	setup	55.51	52.69	44.05	50.76	48.18	51.69	83.08	46.79	61.35	51.32	54.54
access/add	pred.	0.16	0.16	0.16	0.17	0.16	0.16	0.17	0.16	0.16	0.16	0.16
Setvector,	setup	55.51	52.69	44.05	50.76	48.18	51.69	83.08	46.79	61.35	51.32	54.54
access/mul	pred.	0.16	0.16	0.16	0.17	0.16	0.16	0.17	0.16	0.17	0.17	0.16
Setvector,	setup	55.51	52.69	44.05	50.76	48.18	51.69	83.08	46.79	61.35	51.32	54.54
diff./add	pred.	0.16	0.16	0.16	0.17	0.16	0.16	0.17	0.17	0.17	0.16	0.16
Setvector,	setup	55.51	52.69	44.05	50.76	48.18	51.69	83.08	46.79	61.35	51.32	54.54
diff./mul	pred.	0.16	0.17	0.16	0.17	0.16	0.16	0.17	0.17	0.17	0.16	0.17

5 Results

Table 2 shows the accuracy of the evaluated methods and variations, Table 3 shows the execution time of the methods, subdivided into time that has to be spent *offline* (row *setup*), and time that has to be spend *online* (row *pred.*) when calculating the prediction for a specific combination.

Table 2 shows that Chandra's circular sequence based method that utilizes the access frequency on a *per set* basis performs with the highest accuracy ($MRD = 0.58$). However, 676.83 picoseconds have to be spent per instruction (ps/instr.) on average to calculate the predictions with Chandra et al.'s method; this is about 6768 times as long as a prediction takes applying the Activity vector method (0.10 ps/instr.) and about 3981 times as long as a prediction takes applying the Setvector method (0.17 ps/instr.). Although the Activity vector method performs quite fast, it shows a high error rate ($MRD = 3.07$ and $MRD = 2.38$ respectively). However, selecting the higher part of the set bits had been a good idea. Increasing the number of super sets to the number of sets and applying natural numbers to count the number of accesses to each set instead of using only a single bit per set significantly improves accuracy ($MRD = 0.64$, as seen from *Setvector - access, add*), but also increases prediction time (0.16). The Setvector method that applies both access frequency and number of accesses from different keys shows about the same prediction time (0.17 ps/instr.), but a slightly better accuracy ($MRD = 0.60$), that nearly matches that of the about 4000 times slower performing circular sequence based method.

6 Conclusion

In this section, I presented state-of-the art methods to predict cache contention and proposed a new prediction method based on the calculation of so-called *Setvectors*. I simulated the additional memory access time introduced from cache contention during application co-scheduling and compared those values to the prediction methods applying a new metric called *MRD* (mean ranking distance) that calculates the mean difference between the predicted and the simulated ranking. The results showed that the method introduced by Chandra et al. [1] might be the most accurate one, but it is nearly 4000 times slower than the proposed Setvector method, that achieves nearly the same accuracy ($MRD = 0.60$ instead of $MRD = 0.58$).

References

1. Chandra, D., Guo, F., Kim, S., Solihin, Y.: Predicting inter-thread cache contention on a chip multi-processor architecture. In: Proceedings of the 11th Int'l Symposium on High-Performance Computer Architecture (HPCA-11 2005) (2005)
2. Fedorova, A.: Operating system scheduling for chip multithreaded processors. PhD thesis, Harvard University, Cambridge, Massachusetts (2006)
3. Hill, M.D., Smith, A.J.: Evaluating associativity in CPU caches. *IEEE Trans. Comput.* **38**, (1989)

4. Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klausner, A., Lowney, G., Wallace, S., Reddi, V.J., Hazelwood, K.: Pin: Building customized program analysis tools with dynamic instrumentation. In: *Programming Language Design and Implementation* (2005)
5. Mattson, R.L., Gecsei, J., Slutz, D.R., Traiger, I.L.: Evaluation techniques for storage hierarchies. *IBM Syst. J.* **9**, (1970)
6. Settle, A., Kihm, J.L., Janiszewski, A., Connors, D.A.: Architectural support for enhanced SMT job scheduling. In: *Proceedings of the 13th International Conference of Parallel Architectures and Compilation Techniques* (2004)
7. Wulf, W.A., McKee, S.A.: Hitting the memory wall: implications of the obvious. *Comput. Arch. News* **23**(1), (1995)
8. Zwick, M., Obermeier, F., Diepold, K.: Predicting cache contention with setvectors. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 244–251 (2010)
9. Zwick, M., Durkovic, M., Obermeier, F., Diepold, K.: Setvectors for memory phase classification. In: *International Conference on Computer Science and its Applications (ICCSA'09)* (2009)
10. Zwick, M., Durkovic, M., Obermeier, F., Bamberger, W., Diepold, K.: MCCCsim – A highly configurable multi core cache contention simulator. Technical Report – Technische Universität München. <https://mediatum2.ub.tum.de/doc/802638/802638.pdf>, 2009

New Material Model for Describing Large Deformation of Pressure Sensitive Adhesive

Kazuhisa Maeda, Shigenobu Okazawa, and Koji Nishiguchi

Abstract A material model to describe large deformation of pressure sensitive adhesive (PSA) is presented. A relationship between stress and strain of PSA includes viscoelasticity and rubber-elasticity. Therefore, we propose the material model for describing viscoelasticity and rubber-elasticity and formulate the presented material model for finite element analysis. And we validate the present formulation by using one axis tensile calculation.

Keywords Large deformation · Viscoelasticity · Rubber-elasticity · Pressure sensitive adhesive · Material model · Finite element method

1 Introduction

Pressure sensitive adhesive, which is called PSA afterwards, are industrially indispensable products in various fields. Its elastic modulus is about 10^5 Pa at room temperature and indicates extremely low compared with other solid materials. Therefore, large deformation behavior can be observed in conventional PSA deformation.

Figure 1 shows the tensile stress-strain curve of PSA. As shown in Fig. 1, the stress increases exponentially for large strain zone. This behavior is called “rubber-elasticity” in this paper. PSA is generally considered to be a viscoelastic material. However, only viscoelasticity cannot describe practical behavior including rubber-elasticity of PSA consistently [1].

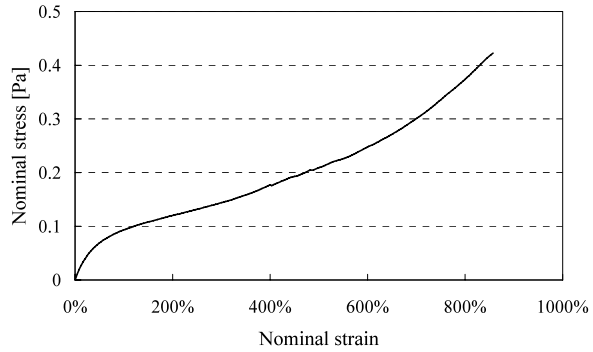
The generalized Maxwell model [2] is usually used for describing viscoelasticity. On the other hands, hyperelasticity is popular to simulate increase in stress [3]. However, there is a difficulty in use of hyperelasticity, because hyperelasticity independent on time. In order to evaluate material parameters, hyperelasticity needs

K. Maeda (✉)

Nitto Denko Corporation, Toyohashi, Aichi, 441-3194, Japan

e-mail: kazuhisa_maeda@gg.nitto.co.jp

Fig. 1 Stress strain curve of PSA



time independent parameters with experimental data without stress relaxation. The aim of this study is the establishment of material model describing visco and rubber elasticity for PSA. The established material model can indicate rubber-elasticity without hyperelastic model. We formulate the above material model and its rate formulation for finite element analysis and then show the validation by using computational example.

2 Material Model

In this section, material model for PSA is described. After introducing generalized Maxwell model, the model is extended to practical model including rubber-elasticity.

2.1 Modification of Generalized Maxwell Model

PSA indicates remarkable viscoelasticity at room temperature. The generalized Maxwell model for describing viscoelasticity is used in the present study. Figure 2 shows generalized Maxwell model. Where i denotes unit number of the generalized Maxwell model. And E and η are elastic modulus of spring and viscous coefficient of dashpot, respectively.

The generalized Maxwell model of Fig. 2 cannot describe rubber-elastic behavior of PSA as shown in Fig. 1. The reason is that elastic modulus of the generalized Maxwell model is constant. Then, we propose evaluation of elastic modulus of the spring component. Figure 3 shows the modified generalized Maxwell model. This proposed model is called “Advanced generalized Maxwell model”. In the advanced generalized Maxwell model, elastic modulus is function of total strain. In addition, viscous coefficients of dashpot are function of strain to assume that relaxation time, τ , is constant like the generalized Maxwell model.

Fig. 2 Generalized Maxwell model

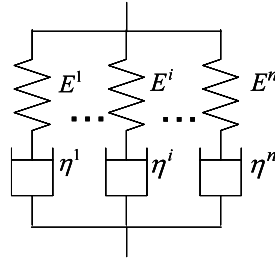
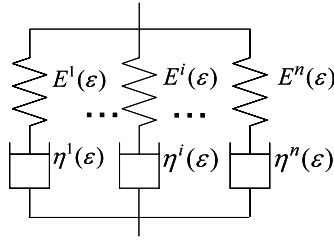


Fig. 3 Advanced generalized Maxwell model



2.2 Elastic Moduli

Elastic moduli for the advanced generalized Maxwell model are determined as the function of total strain. First, we measure stress relaxation behavior with various initial strains in order to investigate the strain dependency of an elastic modulus. For this experiment, standard acrylic PSA is used. Cylindrical PSA sample, whose cross-section is 2 mm^2 , is attached to a tensile machine so that the length of PSA is 10 mm, and initial strain is given 100% by nominal strain. Then, the sample is elongated and keeps in the fixed strain. The stress relaxation curve is obtained by measuring the stress change at the measurement time. The relationship between nominal stress and nominal strain is changed into the relationship between true stress and true strain. Regression analysis is applied to the obtained curve using the stress relaxation formula of the generalized Maxwell model as shown in Eq. 1, and we get the 5 sets of relaxation time, τ^i , and elastic modulus of the spring component, E^i .

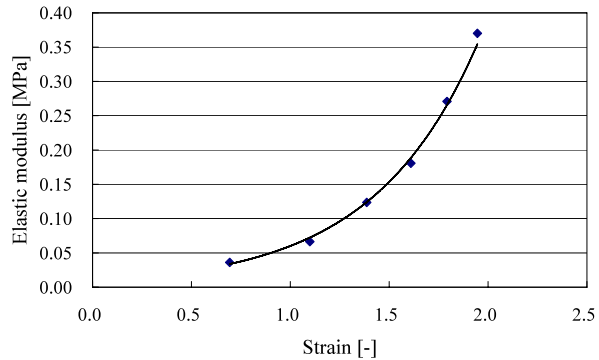
$$\sigma = \sum_{i=1}^n E^i \exp\left(-\frac{t}{\tau^i}\right) \tag{1}$$

where σ , E , τ , and i denote true stress, elastic modulus of a spring component, relaxation time, and the unit number respectively. Then, the stress relaxation measurement with initial strain 200, 300, 400, 500, and 600% is measured, and we get the sets of the relaxation time and elastic modulus at each initial strain.

Table 1 shows relaxation times and elastic moduli of each strain. The strains are converted into true strain. Elastic modulus E is depends on the strain and is considered about each case of τ . The case of $\tau = 100$ [sec] is considered for an example. Figure 4 shows elastic modulus depends on strain.

Table 1 Relaxation times and elastic moduli of each strain

Relation time [s]	True strain					
	0.6931	1.0986	1.3863	1.6094	1.7918	1.9459
1.00×10^0	0.1541	0.3637	0.5694	1.2057	2.0546	3.8663
1.00×10^1	0.0326	0.0832	0.1368	0.3548	0.5107	0.7711
1.00×10^2	0.0363	0.0665	0.1237	0.1808	0.2078	0.3701
1.00×10^3	0.0250	0.0519	0.0761	0.1040	0.1549	0.1522
1.00×10^{10}	0.0639	0.1326	0.2150	0.3279	0.4608	0.5936

Fig. 4 Relationship between elastic modulus and strain

Considering relationship between elastic modulus and strain in Fig. 4. We decide to use the exponential function of strain as a function of an elastic modulus as follows.

$$E^i = A^i \exp(B^i \varepsilon) \quad (2)$$

where A and B denotes material parameter, and ε does strain.

3 Rate Form of Constitutive Equation

Although the elastic modulus of the spring component of the advanced model is defined by Eq. 2, it is necessary to distinguish scalar, vector, and tensor strictly in the case of dealing with three dimensions. So, the elastic modulus of a spring component is replaced with Eq. 3.

$$E^i = A^i \exp(B^i \hat{\varepsilon}) \quad (3)$$

where $\hat{\varepsilon}$ denotes scalar of strain. And $\hat{\varepsilon}$ is defined by Eq. 4 using the strain tensor, $\boldsymbol{\varepsilon}$.

$$\hat{\varepsilon} = \sqrt{\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}} \quad (4)$$

It is assumed that viscoelasticity is in a deviatoric component in the model. So, a constitutive equation of deviatoric and volumetric component is respectively formulated, and then the constitutive equation of whole component is formulated.

3.1 Deviatoric Component

The spring of i th unit is assumed to be an incompressible linear elastic material. The deviatoric stress tensor of i th spring unit is

$$\sigma'^i = 2G^i \mathbf{e}^{sp,i} \quad (5)$$

The superscript prime of a right shoulder shows a deviatoric component. $\mathbf{e}^{sp,i}$ is a small strain tensor of the spring of i th unit. G is a shear elastic modulus, and the relation with E is as follows using Poisson ratio, ν .

$$G^i = \frac{E^i}{2(1 + \nu^i)} \quad (6)$$

The material time derivative of both sides of Eq. 5 gives

$$\frac{D\sigma'^i}{Dt} = 2G^i \mathbf{D}^{sp,i} + \frac{DG^i}{Dt} \frac{\sigma'^i}{G^i} \quad (7)$$

where \mathbf{D} denotes rate of strain tensor. The left side of Eq. 7 does not have objectivity. So the constitutive equation is not objective. Therefore object stress rate is assumed

$$\left(\frac{D\sigma'^i}{Dt} \right)_{(*)} = 2G^i \mathbf{D}^{sp,i} + \frac{DG^i}{Dt} \frac{\sigma'^i}{G^i} \quad (8)$$

where subscript $(*)$ means arbitrary objective stress rate. Equation 8 is used as a constitutive equation of a spring.

Next, the dashpot of i th unit is considered. The shear viscous coefficient and the strain rate tensor of i th dashpot unit is expressed as η^i and $\mathbf{D}^{dp,i}$ respectively. And dashpot is assumed to be incompressible Newtonian fluid. The constitutive equation of i th dashpot unit is given as

$$\sigma'^i = 2\eta^i \mathbf{D}^{dp,i} \quad (9)$$

The model property insists that the strain tensor of i th unit can be assumed to be equal to the strain of the whole model. That is, $\varepsilon = \varepsilon^{sp,i} + \varepsilon^{dp,i}$. The material time derivative of this equation gives

$$\mathbf{D} = \mathbf{D}^{sp,i} + \mathbf{D}^{dp,i} \quad (10)$$

From Eqs. 8, 9, 10, the constitutive equation of the i th unit's deviatoric component is derived to

$$\left(\frac{D\sigma'^i}{Dt} \right)_{(*)} = 2G^i \mathbf{D} + \frac{DG^i}{Dt} \frac{\sigma'^i}{G^i} - \frac{\sigma'^i}{2(1 + \nu^i)\tau^i} \quad (11)$$

where ν is Poisson ratio. Equation 3 can be changed to

$$\frac{DE^i}{Dt} = B^i E^i \frac{D\hat{\varepsilon}}{Dt} \quad (12)$$

And Eq. 11 becomes

$$\left(\frac{D\sigma'^i}{Dt} \right)_{(*)} = 2G^i \mathbf{D} + B^i \frac{D\hat{\varepsilon}}{Dt} \frac{\sigma'^i}{G^i} - \frac{\sigma'^i}{2(1 + \nu^i)\tau^i} \quad (13)$$

Since the stress of the whole model is derived from summation of the stress of each unit, the constitutive equation of the whole model is expressed to

$$\left(\frac{D\sigma'^i}{Dt}\right)_{(*)} = \sum_i \left(2G^i \mathbf{D} + B^i \frac{D\hat{\varepsilon}}{Dt} \sigma'^i - \frac{\sigma'^i}{2(1+\nu^i)\tau^i} \right) \quad (14)$$

3.2 Volumetric Component

Here it is assumed that the volumetric component is a compressible linear elastic material. Pressure, p , is given as

$$p = -K_v \operatorname{tr} \boldsymbol{\varepsilon} \quad (15)$$

where K_v is defined as follows

$$K_v = \sum_{i=1}^n \left\{ \frac{E^i}{3(1-2\nu^i)} \right\} \quad (16)$$

The material time derivative of Eq. 15 gives

$$\frac{Dp}{Dt} = \frac{DK_v}{Dt} \frac{p}{K_v} - K_v \operatorname{tr} \mathbf{D} \quad (17)$$

where

$$\frac{DK_v}{Dt} = \sum_{i=1}^n \left\{ \frac{1}{3(1-2\nu^i)} \frac{DE^i}{Dt} \right\} \quad (18)$$

Substituting Eq. 12 into Eq. 18 gives

$$\frac{DK_v}{Dt} = \sum_{i=1}^n \left\{ \frac{B^i E^i}{3(1-2\nu^i)} \frac{D\hat{\varepsilon}}{Dt} \right\} \quad (19)$$

Therefore, the constitutive equation of volumetric component is

$$\frac{Dp}{Dt} = \frac{p}{K_v} \sum_{i=1}^n \left\{ \frac{B^i E^i}{3(1-2\nu^i)} \frac{D\hat{\varepsilon}}{Dt} \right\} - K_v \operatorname{tr} \mathbf{D} \quad (20)$$

3.3 Whole Component

Here the Jaumann rate is selected for objective stress rate. The material time derivative of Cauchy stress and its Jaumann rate are related in Eq. 21.

$$\left(\frac{D\boldsymbol{\sigma}}{Dt}\right)_{(J)} = \frac{D\boldsymbol{\sigma}}{Dt} + \mathbf{W} \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \mathbf{W} \quad (21)$$

where \mathbf{W} denotes spin tensor, and the lower right (J) shows the Jaumann rate. Here the Cauchy stress is divided into deviatoric and volumetric components,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}' - p\mathbf{I} \quad (22)$$

where \mathbf{I} denotes 2nd unit tensor. Substituting Eq. 22 into Eq. 21 gives

$$\left(\frac{D\boldsymbol{\sigma}}{Dt}\right)_{(J)} = \left(\frac{D\boldsymbol{\sigma}'}{Dt}\right)_{(J)} - \frac{Dp}{Dt}\mathbf{I} \quad (23)$$

Therefore, substituting constitutive equation of deviatoric and volumetric component into Eq. 23, the constitutive equation of the whole component is,

$$\begin{aligned} \left(\frac{D\boldsymbol{\sigma}^i}{Dt}\right)_{[J]} = & \sum_{i=1}^n \left\{ 2G^i \mathbf{D} + B^i \frac{D\hat{\boldsymbol{\varepsilon}}}{Dt} \boldsymbol{\sigma}'^i - \frac{\boldsymbol{\sigma}'^i}{2(1+\nu^i)\tau^i} \right\} \\ & - \left[\frac{p}{K_v} \sum_{i=1}^n \left\{ \frac{B^i E^i}{3(1-2\nu^i)} \frac{D\hat{\boldsymbol{\varepsilon}}}{Dt} \right\} - K_v \text{tr} \mathbf{D} \right] \mathbf{I} \end{aligned} \quad (24)$$

4 Explicit Finite Element Method

The present study employs an explicit finite element method [4] to calculate the following computational example. The explicit finite element method is computationally robust because of no iterations.

4.1 Discrete Equilibrium Equation

The equilibrium equation ignoring the body force is,

$$\rho \mathbf{a} = \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{x}} \quad (25)$$

where ρ is the material density, \mathbf{a} is the spatial acceleration, and $\boldsymbol{\sigma}$ is the Cauchy stress.

We can derive the virtual work equation by multiplying both sides of Eq. 25 by the arbitrary virtual displacement $\delta \mathbf{u}$ with the Gauss' divergence theorem of volume V .

$$\int_V \rho \mathbf{a} \cdot \delta \mathbf{u} dV + \int_V \boldsymbol{\sigma} : (\delta \boldsymbol{\varepsilon}) dV = \int_{\partial V} \bar{\mathbf{t}} \cdot \delta \mathbf{u} dS \quad (26)$$

where $\bar{\mathbf{t}}$ is the external surface force on the boundary area ∂V , and $\boldsymbol{\varepsilon}$ is the linear strain as follows,

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left[\left(\frac{\partial \mathbf{u}}{\partial x} \right) + \left(\frac{\partial \mathbf{u}}{\partial x} \right)^T \right] \quad (27)$$

The discrete equilibrium equation can be derived using the finite element as follows;

$$\mathbf{M}\mathbf{a} + \mathbf{F}_{\text{int}} = \mathbf{F}_{\text{ext}} \quad (28)$$

where \mathbf{M} is the mass matrix, \mathbf{F}_{int} and \mathbf{F}_{ext} are the internal and external force vectors respectively. For the numerical integration of the isoparametric element in the plane strain state, the selective reduced integration method is used to avoid volumetric locking [5].

4.2 Central Difference Method

To advance the time of the discrete equilibrium Eq. 28, we select the central difference method. Here, Δt is the time increment from time t^n to t^{n+1} . The current time is t^n and any properties of the material at time t^{n+1} will be explicitly calculated with the central difference method. The material coordinate \mathbf{x} at t^{n+1} is evaluated with the material velocity \mathbf{v} at the central incremental time $t^{n+1/2}$.

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t \quad (29)$$

where the material velocity \mathbf{v} at time $t^{n+1/2}$ is

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a}^n \Delta t \quad (30)$$

and the spatial acceleration \mathbf{a} at time t^n is solved as follows with Eq. 28.

$$\mathbf{a}^n = \mathbf{M}^{-1} (\mathbf{F}_{\text{ext}}^n - \mathbf{F}_{\text{int}}^n) \quad (31)$$

Equation 31 requires no solution of the simultaneous equations by using the diagonal lumped mass matrix for \mathbf{M} .

5 Computational Results

Here, in order to verify an above-mentioned technique, one axis tensile deformation of PSA is analyzed.

5.1 Material Constants

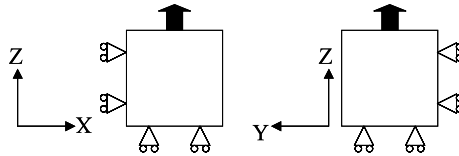
The material constants which must be defined in the constitutive equation are A^i , B^i and τ^i . These values are calculated from the experimental data of one axis elongation measurement.

First, the method of one axis elongation measurement is explained. It is measured at room temperature. Cylindrical PSA sample whose cross-section is 2 mm² is attached to the tensile machine so that the length of PSA is 10 mm, the sample is

Table 2 Material constants

Relaxation times [s]	A^i [Pa]	B^i
1.00×10^0	1.28×10^5	1.17
1.00×10^1	8.93×10^1	4.40
1.00×10^2	3.71×10^2	3.67
1.00×10^3	1.30×10^5	0.67
1.00×10^6	7.31×11	3.85

Fig. 5 Boundary conditions



elongated with constant speed. The speed is 10, 50, 300 mm/min. Since the measurement data is nominal stress and nominal strain, the stress-strain curve is changed into true stress-true strain.

The constitutive equation of the advanced model calculated by one dimension is

$$\frac{d\sigma}{dt} = \sum A^i \exp(B^i \varepsilon) - \sum \left(\frac{1}{\tau^i} - B^i \frac{d\varepsilon}{dt} \right) \sigma^i \tag{32}$$

Equation 32 is applied to the stress-strain curve with nonlinear least squares method, and the material constants are determined. Approximate curve is calculated so that all stress-strain curves with different three elongation speed are satisfied. The result is shown in Table 2.

5.2 Comparison with Experiment

In order to verify the proposed constitutive equation, computational results is compared with experiments. The stress-strain analysis of PSA is treated in order to examine the quantity with experimental results. Figure 5 shows analysis model and boundary conditions with uniform deformation. The analysis object is cubic PSA whose length of one side is 1 cm. The number of elements of the material model as shown in Fig. 3 is set to 5, and material parameters use the values of Table 2. The density of PSA uses 1000 kg/m³. In order to assume near incompressibility, the Poisson ratio in all the elements of the material model is set to 0.49.

The aim of this paper is to develop the constitutive equation of a PSA. So, the analysis is calculated by finite element analysis with one lattice. This analysis employs the Lagrangian formulation, which is usually used for solid analysis. The Lagrangian computational mesh is follows material.

The tensile speed is 5, 50, 500 mm/min, which intentionally differ from the rates used at identifying the material parameters. Figure 6 shows the comparison

Fig. 6 Comparison between experiment and computational result in tensile speed 5 mm/min

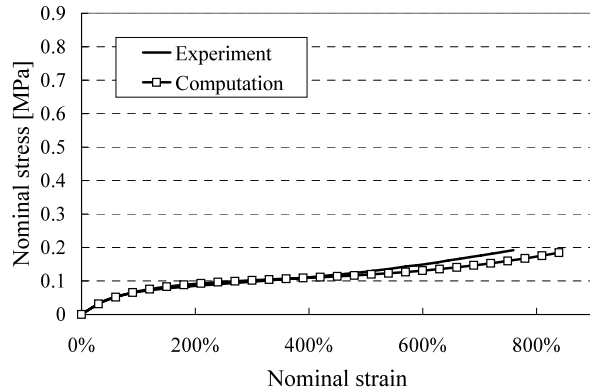
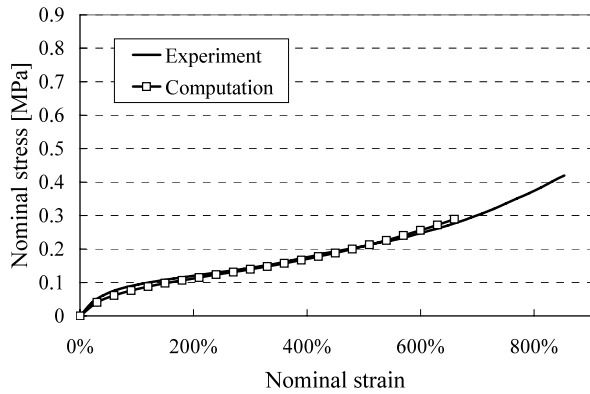


Fig. 7 Comparison between experiment and computational result in tensile speed 50 mm/min



of an experimental and computational result at elongating speed 5 mm/min. The computational result is well consisted with the experiment.

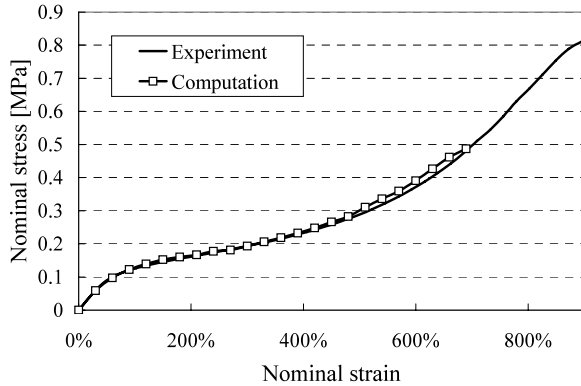
Figures 7 and 8 show the comparison of an experimental and computational result at tensile speed 50 and 500 mm/min respectively. Both show good agreement between experimental and computational results. All results show that computational results can describe increase in stress for rubber elasticity. Furthermore, they show that computational results can describe the rate dependability relevant to the viscoelasticity, which is another PSA's property.

6 Conclusions and Future Works

This paper has described the material model, which can describe the deformation of PSA. Our results indicate the following.

1. The present model, advanced generalized Maxwell model, can describe visco and rubber elasticity.

Fig. 8 Comparison between experiment and computational result in tensile speed 500 mm/min



2. We have formulized the three-dimensional constitutive equation of the advanced generalized Maxwell model.
3. We have validated the proposed advanced generalized Maxwell model with the one axis tensile analysis.

The remained subjects to simulate practical PSA behavior with large deformation are as follows.

1. The present analysis uses dynamic explicit method. Therefore, computational time step size is extremely small because of the requirement of the Courant condition. It is necessary to consider the solution method, which can use large time step size, for suitable analysis of PSA deformation.
2. The highly distorted Lagrangian finite elements cannot retain numerical accuracy. The present formulation should be extended to an Eulerian formulation [6, 7], which is attractive for large deformation problem like PSA.

References

1. Maeda, K., Okazawa, S., Nishiguchi, K.: Visco-rubber elastic model for pressure sensitive adhesive. In: Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineering and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010, pp. 393–398 (2010)
2. Holzapfel, G.A.: Nonlinear Solid Mechanics. Wiley, New York (2001)
3. Simo, J.C., Hughes, T.J.R.: Computational Inelasticity. Springer, Berlin (1973)
4. Okazawa, S., Kashiya, K., Kaneko, K.: Large deformation dynamic solid analysis by Eulerian solution based on stabilized finite element method. *Int. J. Numer. Meth. Eng.* **72**, 1544–1559 (2007)
5. Hughes, T.J.R.: Generalization of selective integration procedures to anisotropic and nonlinear media. *Int. J. Numer. Meth. Eng.* **15**, 1413–1418 (1980)
6. Benson, D.J.: Computational method in Lagrange and Eulerian hydrocodes. *Comput. Methods Appl. Mech. Eng.* **99**, 235–394 (1992)
7. Benson, D.J., Okazawa, S.: Contact in a multimaterial Eulerian finite element formulation. *Comput. Methods Appl. Mech. Eng.* **193**, 4277–4298 (2004)

QoS Provisioning in EPON Systems with Interleaved Two Phase Polling-Based DBA

I-Shyan Hwang, Jhong-Yue Lee, and Zen-Der Shyu

Abstract Ethernet Passive Optical Networks (EPONs) are designed to deliver multiple services and applications, such as voice communications, standard (SDTV) and high-definition video (HDTV). To support these applications and their various requirements, EPONs require Quality-of-Service (QoS) mechanisms to be built in service. For this purpose, a scalable Interleaved Dynamic Bandwidth Allocation (IDBA) mechanism for sharing the uplink bandwidth among optical network units (ONUs) is proposed in this paper. The modus operandi of IDBA is to divide the cycle time by partitioning the ONUs into two groups with some timing overlap to execute interleaved bandwidth allocation, which cooperates with Limited Bandwidth Allocation (LBA), Excess Bandwidth Reallocation (EBR) and accurate prediction mechanism in EPONs. The proposed IDBA mechanism has two advantages, namely it eliminates the idle period problem in the traditional DBA mechanism, and guarantees QoS services by dynamically adjusting the bandwidth within the group of subscribers. This will not only support the differentiated services architecture but also offer various QoS levels. Simulation results obtained show that the proposed IDBA mechanism achieves desirable system performance relative to packet delay, jitter performance, throughput, ratio of packet loss and fairness.

Keywords EPON · QoS · IDBA · System performance · EBR

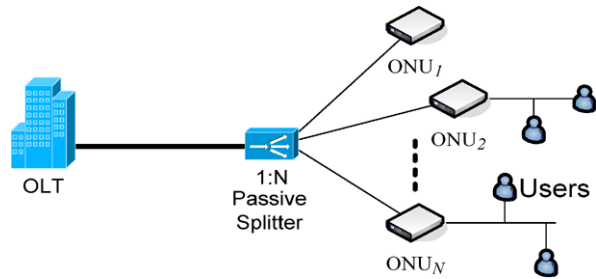
1 Introduction

Due to the rapid and consistent increase in network traffic generated by domestic and small business users over the last few years, broadband access networks have become increasingly important. However, though various technologies have been

I.-S. Hwang (✉)

Department of Computer Science and Engineering, Yuan-Ze University, Chung-Li 32003, Taiwan
e-mail: ishwang@saturn.yzu.edu.tw

Fig. 1 Tree-based PON topology



used to provide broadband access to networks in the area known as the “last mile” [1], they cannot simultaneously upgrade the current access network whilst providing a low-cost and high-speed solution for broadband access services. One possible solution is Ethernet Passive Optical Network (EPON) that has been discussed in IEEE 802.3ah as one of the extensions of Gigabit-Ethernet [2]. The EPON architecture, as shown in Fig. 1, consists of a centralized optical line terminal (OLT) and a number of splitters. It connects a group of associated optical network units (ONUs) over point-to-multipoint topologies to deliver broadband packets and reduces costs relative to maintenance power. EPON also provides bi-directional transmissions, i.e. downstream transmission from OLT to ONUs and upstream transmission from ONUs to OLT in sequence. To avoid data collision, the multi-point control protocol (MPCP) is introduced on EPON, and two media access control (MAC) messages: GATE and REPORT messages are included in MPCP protocol [3].

In the traditional DBA scheme, the OLT will begin bandwidth allocation after collecting whole REPORT messages, resulting in the *idle period problem* (as shown in Fig. 2). To elaborate, the idle period is the sum of the computation time of DBA and the round trip time between OLT and each ONU. Hence, for the DBA scheme, reducing the idle period becomes one of the important issues to address in order to improve bandwidth utilization. Another problem with the DBA scheme is that the queue state is inconsistent due to packets that continue to arrive during this waiting time. To clarify this problem, refer Fig. 3 illustrates the *waiting time*, $t_2 - t_1$, which is the time elapsed from when the packets begin to arrive to before the start of data transmission. Consequently, packets that arrive during the waiting time have to be delayed to the next transmission cycle, potentially leading to longer delays. To address this, predictive schemes can be used so that traffic arrival during the waiting time is taken into consideration to avoid longer packet delay and network performance degradation.

In this paper, we discuss the precision of an accurate traffic prediction mechanism, which is necessary to avoid over-estimation or under-estimation that can result in longer packet delays and degrade the network performance [4–7]. Although exhaustive queue size prediction mechanism have been proposed (which can be credit-based [4, 8, 9], linear-based [4, 5, 10], proportion-based [5, 11], waited-based [12] or QoS-based [6–8, 13, 14]), these traffic prediction mechanism are unable to provide feasible solutions for differentiated services and have not addressed the queue

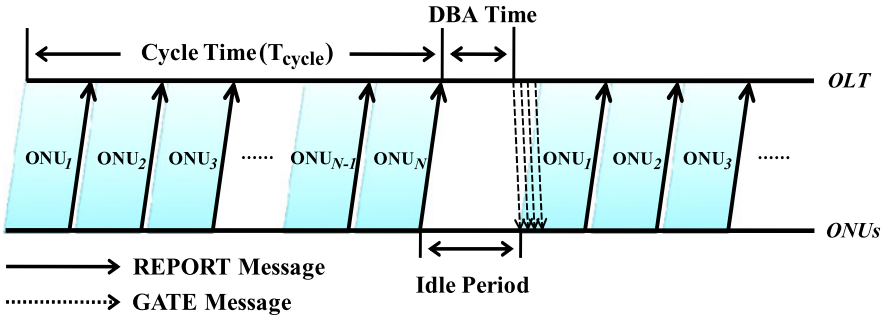


Fig. 2 Traditional DBA mechanism

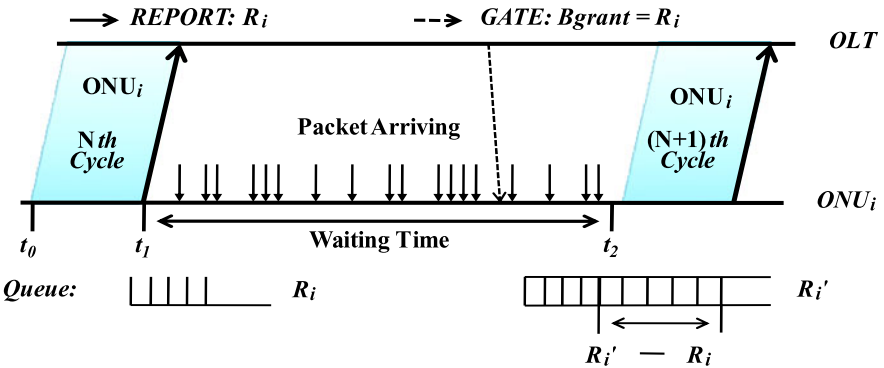


Fig. 3 Queue state between waiting time

size inconsistent problem. In this paper, the proposed traffic prediction mechanism supports differentiated services with their various requirements.

Previous researches have suggested that the maximum of T_{cycle} is 1 ms [15, 16], which is set to meet the ITU-T recommendation, G.114, i.e. the delay for voice traffic in the access network to be set at 1.5 ms [17]. On one hand, making T_{cycle} too large will lead to longer packet delays for all Ethernet frames because a larger cycle time incurs a larger transmission window size and results in the ONU ineffectively holding the transmission channel. As a result, the backlogged traffic at the next ONU experiences longer packet delays. On the other hand, making T_{cycle} too small will result in more bandwidth being wasted by guard intervals and an increase in CPU processing load.

From our previous studies [5], we noted that the idle period problem of IPACT can be resolved by using Early-DBA mechanism with Prediction-based Fair Excessive Bandwidth Reallocation Scheme (PFEBR), which includes the unstable degree list to provide more accurate predictions. However, the Early-DBA has a jitter performance problem, which is due to a change in the transmission orders of some ONUs. A scalable Interleaved Dynamic Bandwidth Allocation (IDBA) mechanism is proposed in this paper, which uses the concept of Early-DBA to resolve the idle

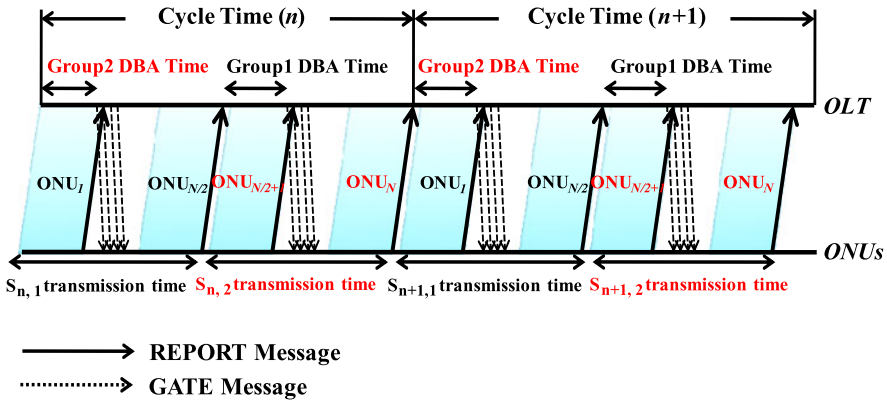


Fig. 4 Operation of the proposed IDBA mechanism

period problem. Nevertheless, the proposed IDBA does not change the granting order for ONUs to reduce the jitter problem.

In proposed IDBA mechanism, shown in Fig. 4, the T_{cycle} is divided by halving the ONUs. One part is the first subgroup (*Group1*) of ONUs, which is denoted by $S_{n,1}$ transmission time for cycle n , and the other part is the second subgroup (*Group2*) of ONUs ($S_{n,2}$). The subgroup $S_{n+1,1}$ upstream transmission period is calculated in the n th cycle. At *Group2* DBA time, the OLT performs the DBA computation for ONUs in subgroup $S_{n,2}$. At this time, the OLT has granted the GATE message to ONUs in subgroup $S_{n+1,1}$, so that the ONUs in subgroup $S_{n+1,1}$ can transmit upstream data during idle time while the OLT computes DBA for subgroup $S_{n+1,2}$ (*Group2* DBA). ONUs in subgroup $S_{n+1,2}$ is allowed to transmit upstream data as soon as the last ONU in subgroup $S_{n+1,1}$ finishes transmission. The OLT lets the DBA process execute the QoS-based prediction, the limit bandwidth allocation (LBA), and the excessive bandwidth reallocation (EBR) for each part. When the predicted bandwidth has been over-estimated, the unused bandwidth is simply reserved for the next part, and the total transmission time of two successive parts is limited in one T_{cycle} . The proposed IDBA has two contributions: one is eliminating the idle period problem in the traditional DBA mechanism, and the other is ensuring QoS services by dynamically adjusting the bandwidth between $S_{n,1}$ and $S_{n,2}$, which not only supports differentiated services architecture but also offers various levels of QoS.

The rest of this paper is organized as follows. Section 2 presents the proposed interleaved DBA and QoS-based scheduling scheme. Simulation evaluations are presented in Sect. 3, and conclusions are drawn in Sect. 4.

2 Proposed Interleaved DBA Mechanism

The Interleaved Dynamic Bandwidth Allocation (IDBA) mechanism is proposed to resolve the *idle period problem* and enhance the QoS for differentiated services by

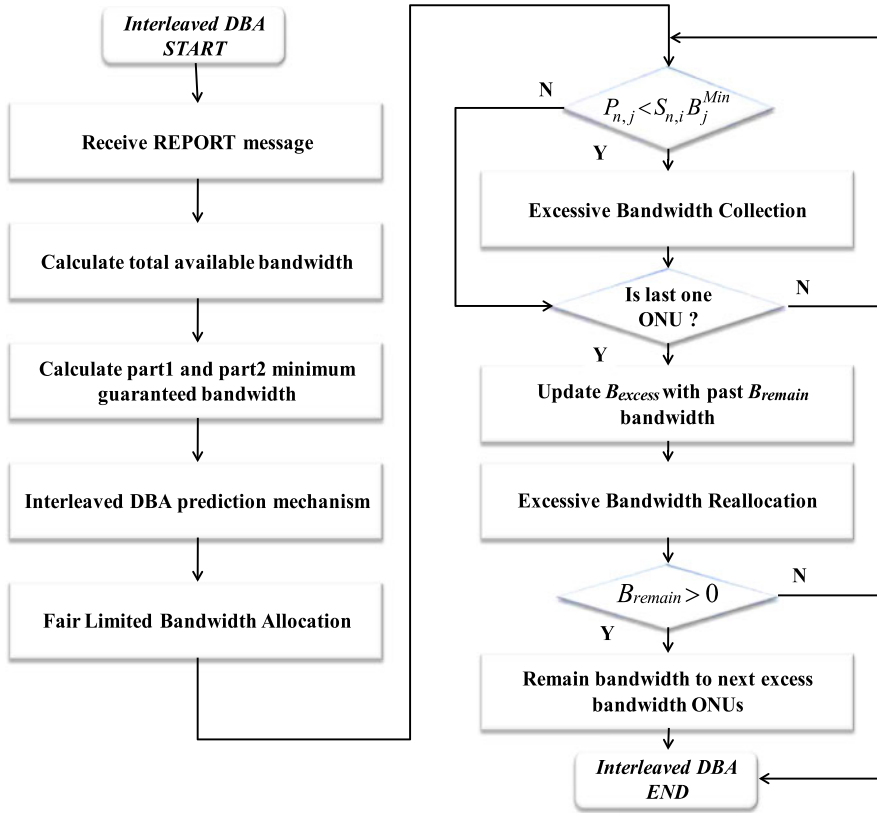


Fig. 5 Flowchart of IDBA mechanism

using prediction and EBR in the EPON system. The flowchart of the IDBA mechanism is illustrated in Fig. 5, where $P_{n,j}$ is defined as the bandwidth request prediction of ONU_j for cycle n , $S_{n,i} B_j^{Min}$ is defined as the minimum guaranteed bandwidth belonging to part i of ONU_j for cycle n , which can be calculated by the service level agreement (SLA), B_{excess} the excess bandwidth which is calculated by the sum of the under exploited bandwidth of lightly-loaded ONUs and B_{remain} , the unused bandwidth after excess bandwidth reallocation from heavily-loaded ONUs.

When receiving whole REPORT messages from each ONU, the total available bandwidth can be calculated as $(r \times (T_{cycle}^{Max} - N \times T_g) - N \times 512)$, where r is the transmission speed of the EPON in bits per second, T_{cycle}^{Max} is the maximum cycle time, N is the number of ONUs, T_g is the guard time and the control message length is 512 bits for the EPON system. Next, the QoS-based IDBA executes the prediction mechanism based on the current traffic status. The limited bandwidth allocation mechanism then compares the minimum guaranteed bandwidth with the predicted bandwidth of each ONU. If $P_{n,j} \leq S_{n,i} B_j^{Min}$, then excessive bandwidth collection for lightly-loaded ONUs is executed, followed by excessive bandwidth

reallocation mechanism for heavily-loaded ONUs. In the end, the unused bandwidth from the over-estimated bandwidth can be reserved for the next group of ONUs for DBA. Therefore, the IDBA can support QoS and enhance system performance for differential services and efficiently reallocates excessive bandwidth in EPON. The prediction mechanism of IDBA is described in Sect. 2.1, LBA mechanism with fairness in Sect. 2.2 and EBR mechanism in Sect. 2.3 respectively.

Initially, the available bandwidth for part one, $S_{n,1}B_{available}$, and part two, $S_{n,2}B_{available}$, for cycle n , can be calculated by applying Eqs. 1 and 2.

$$S_{n,1}B_{available} = total\ available\ bandwidth \times \left(1 - \sum_{j=N/2+1}^N S_{n,2}W_j \right), \quad (1)$$

$$S_{n,2}B_{available} = total\ available\ bandwidth \times \left(1 - \sum_{j=1}^{N/2} S_{n,1}W_j \right), \quad (2)$$

where W_j is the weight assigned to each ONU $_j$ based on its SLA.

2.1 Interleaved DBA Prediction Mechanism

In relation to resolving queue variation between waiting times and reducing the packet delay, the prediction mechanism of IDBA takes differential traffic characteristic into account to enhance the prediction accuracy for each ONU. In this paper, we divide traffic data into three priority classes, EF, AF, and BE by the definition of Differentiated Services [4, 18]. To achieve better performance for a time-critical application, for instance constant bit rate (CBR) for EF traffic and non-busy traffic mode, bandwidth should be assigned to the ONUs according to the rate of these applications. Therefore, the proposed prediction mechanism assigns the CBR bandwidth to EF traffic as it multiplies the previous request of EF by one plus the proportion of waiting time, $T_{waiting,j}$, and cycle time, $T_{cycle,j}$, for ONU $_j$. Moreover, the traffic characteristic of AF and BE are variable bite rate (VBR) and busy traffic mode, and the proposed prediction mechanisms of AF and BE traffic compare the difference between the requested transmission window at the present cycle and a mean value requested transmission window of historical cycles. The predicted value of bandwidth requirements for differentiated traffic is expressed in Eq. 3, where $R_{n,j}^T$ represents bandwidth request of each traffic type of ONU $_j$ in cycle n , and $\overline{H_j^T}$ is the average bandwidth requirements of the history cycle of each traffic type of ONU $_j$, where $T \in \{EF, AF, BE\}$.

$$\begin{cases} P_{n,j}^{EF} = \overline{H_j^{EF}} \times (1 + T_{waiting,j}/T_{cycle,j}) \\ P_{n,j}^{AF} = R_{n,j}^{AF} - \overline{H_j^{AF}} \\ P_{n,j}^{BE} = R_{n,j}^{BE} - \overline{H_j^{BE}}. \end{cases} \quad (3)$$

After the traffic forecast value is calculated in each ONU, the prediction mechanism can derive $P_{n,j}^T$ which represents the prediction value of each ONU $_j$ in cycle n , where $T \in \{EF, AF, BE\}$. For AF and BE traffics, if $P_{n,j}^T > 0$, the demand tends to increase gradually and so we update the forecast value to obtain the new bandwidth requirements. Otherwise, if $P_{n,j}^T \leq 0$, we do not update.

2.2 Fair Limited Bandwidth Allocation Mechanism

During dynamic allocation, the allocated timeslot will be adapted to the requested bandwidth. To prevent the allocation of excessive bandwidth (which can result in wasted bandwidth) or not enough bandwidth (which can increase packet delay), the proposed LBA is set as $S_{n,i}G_j = \min(P_{n,j}, S_{n,i}B_j^{Min})$, where $S_{n,i}G_j$ is the granted bandwidth timeslot in GATE message for ONU $_j$ (which belongs to the i th part in the n th cycle), $P_{n,j}$ is the predicted value of bandwidth requirements for ONU $_j$, and $S_{n,i}B_j^{Min}$ is the minimum guaranteed bandwidth of ONU $_j$ that will equalize the bandwidth for each ONUs in part i . If $P_{n,j} < S_{n,i}B_j^{Min}$, the limited granted bandwidth from the OLT is the same as the predicted bandwidth; otherwise, the grant for the ONU $_j$ equals $S_{n,i}B_j^{Min}$. The proposed LBA not only solves the problem of an ONU with heavy traffic load monopolizing the upstream channel, but also supports the priority servicing of differentiated services to guarantee QoS. The $S_{n,i}B_j^{Min}$ and $S_{n,i}G_j^T$, where $T \in \{EF, AF, BE\}$, in GATE message based on each traffic class are described as follows:

$$S_{n,i}B_j^{Min} = S_{n,i}B_{available}/Numbers\ of\ S_{n,i}ONU, \quad (4)$$

$$\begin{cases} S_{n,i}G_j^{EF} = \min(P_j^{EF}, S_{n,i}B_j^{Min}) \\ S_{n,i}G_j^{AF} = \min(P_j^{AF}, S_{n,i}B_j^{Min} - S_{n,i}G_{j,n+1}^{EF}) \\ S_{n,i}G_j^{BE} = S_{n,i}B_j^{Min} - S_{n,i}G_j^{EF} - S_{n,i}G_j^{AF}. \end{cases} \quad (5)$$

2.3 Excessive Bandwidth Reallocation Mechanism

After LBA grants all bandwidth timeslot to the active ONU $_j$, lightly loaded ONUs with bandwidth requirements less than the $S_{n,i}B_j^{Min}$ may still be present. The sum of the under utilized bandwidth of lightly-loaded ONUs, excessive bandwidth (B_{excess}) [5, 6], can be expressed as Eq. 6:

$$B_{excess} = \sum_{j \in L} (S_{n,i}B_j^{Min} - P_{n,j}), \quad (6)$$

where $S_{n,i}B_j^{Min} > P_{n,j}$, L is the set of lightly-loaded ONUs and j is a lightly-loaded ONU in L .

In the proposed EBR mechanism, B_{excess} is redistributed among the heavily-loaded ONUs. A heavily-loaded ONU obtains an additional bandwidth based on the EBR mechanism. After EBR, if the B_{excess} is larger than the sum of the bandwidth request among the heavily-loaded ONUs, then a remaining available bandwidth, B_{remain} , which can be retained to the next excessive bandwidth collection to enhance the bandwidth efficiency in the next cycle. The B_{remain} must be restricted at half cycle time to avoid the piling up of unused available bandwidth, which can result in unfair resource distribution. B_{remain} is expressed in Eq. 7 as follows:

$$B_{remain} = B_{excess} - \sum_{j \in H} (P_{n,j} - S_{n,i} B_j^{Min}), \quad (7)$$

where $S_{n,i} B_j^{Min} < P_{n,j}$, H is the set of heavily-loaded ONUs and j is a heavily-loaded ONU in H .

The proposed EBR mechanism in the IDBA can provide fairness to EBR based on the guaranteed bandwidth rather than the requested bandwidth, with no partiality and increase in bandwidth utilization. Moreover, the EBR mechanism not only alleviates the unfairness problem but also supports QoS by fairly distributing in a priority manner to improve LBA scheduling and enhance the traffic class.

3 Performance Evaluation

In this section, comparisons are performed using the PFEBR scheme, IDBA_Fixed without LBA, EBR and prediction, IDBA_EBR without prediction mechanism, IDBA_EBR_Pre incorporated QoS-based prediction with EBR, hybrid double-phase polling algorithm (DPA) [8], with respect to end-to-end delay, throughput, jitter performance, ratio of packet loss and fairness. The system model is set up within the OPNET simulator with one OLT and 32 ONUs. The simulation scenario is summarized in Table 1. For the traffic model considered, an extensive study has shown that most network traffic can be characterized by self-similarity and long-range dependence (LRD) [19]. In order to simulate the effect of high priority traffic, the proportion of traffic profile is analyzed by simulating the three significant scenarios in (EF, AF, and BE) with (20%, 40%, 40%), (40%, 30%, 30%), and (60%, 20%, 20%), respectively [6, 20].

3.1 End-to-End Packet Delay

Figure 6 compares the mean end-to-end packet delay and EF traffic classes with end-to-end delay vs. different traffic loads for PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA. The simulation results obtained show that hybrid DPA has a higher average end-to-end packet delay than IDBA when the traffic load exceeds 60%. However, the mean end-to-end packet delay of IDBA_Fixed and IDBA_EBR_Pre increased when traffic load exceeded 70% while for IDBA_EBR,

Table 1 Simulation scenario

Number of ONUs in the system	32
Upstream/downstream link capacity	1 Gbps
OLT-ONU distance (uniform)	10–20 km
Buffer size	10 MB
Maximum transmission cycle time	1 ms
Guard time	5 μ s
Computation time of DBA	10 μ s
Control message length	64 bytes

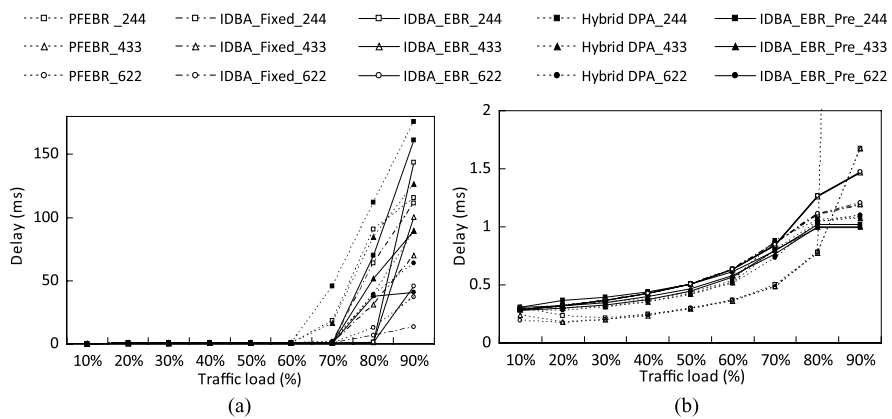


Fig. 6 (a) Average end-to-end packet delay and (b) EF end-to-end delay vs. different traffic loads for PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA

it increased when traffic load exceeded 80% regardless of the scenario. One possible reason is that the *idle period problem* can be resolved by IDBA mechanism and that the excess bandwidth can be reallocated effectively by using the EBR mechanism. Figure 6(a) shows that the IDBA mechanism has better performance in terms of mean end-to-end packet delay when the ratio of EF traffic increases. One possible reason is that the network environment will converge to a stable state when the proportion of CBR traffic is higher. Figure 6(b) shows that the IDBA_EBR_Pre can guarantee the bandwidth for EF traffic class, and result in lower EF end-to-end delay for each ONU. The IDBA mechanism meets the ITU-T recommendation G.114 that specifies the delay for voice traffic in the access network at 1.5 ms [17]. The end-to-end delay of EF traffic in the proposed method can be guaranteed regardless of the proportion of EF traffic.

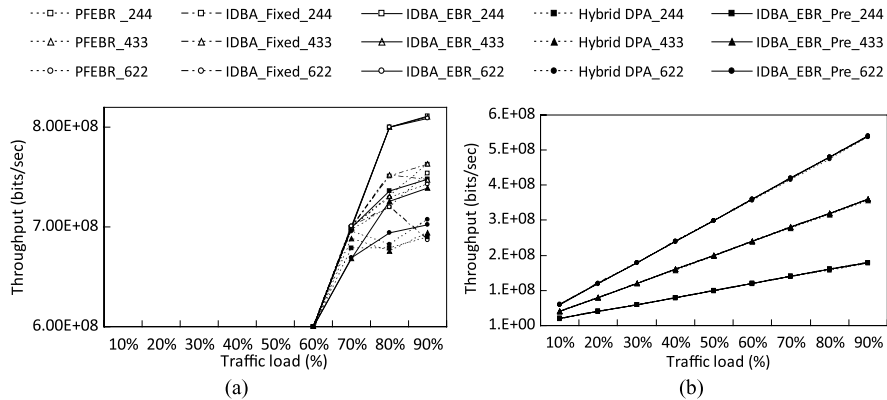


Fig. 7 (a) Mean system throughput and (b) EF traffic throughput vs. different traffic loads for PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA

3.2 System Throughput

Figure 7 shows the mean system throughput against different traffic loads for the PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA. Figure 7(a) shows that the proposed IDBA mechanism outperforms hybrid DPA in mean system throughput, and this is because interleaved transmissions can eliminate the problem of idle time in traditional DBA and support efficient EBR as well as the interleaved remain compensation mechanism. In this case, the IDBA_EBR has the best mean throughput performance due to effective excess bandwidth reallocation and the interleaved remain compensation mechanism. As for the EF throughput, as shown in Fig. 7(b), the proposed IDBA_EBR_Pre mechanism outperforms the other mechanisms because the EF traffic obtains additional bandwidth by using QoS-based prediction mechanism which can enhance high priority traffic adaptively for different traffic proportions.

3.3 EF Jitter and Packet Loss Ratio

Figure 8 shows the comparison of delay variance of EF class and packet loss against different traffic loads among PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA, respectively. In the proposed IDBA mechanism, we can see that the EF jitter of PFEBR can be improved by using IDBA, especially IDBA_Fixed. The reason is that the transmission order of each ONU is sequential and that PFEBR changes the transmission order of ONUs. Figure 8(b) shows that the hybrid DPA begins to have packet loss when the traffic load exceeds 70% in every scenario due to over allocation of the requested bandwidth to ONUs [21] for the EBR mechanism in hybrid DPA. This is termed the *redundant bandwidth problem* [5] to decrease overall system throughput. IDBA_Fixed starts to have packet loss when traffic load

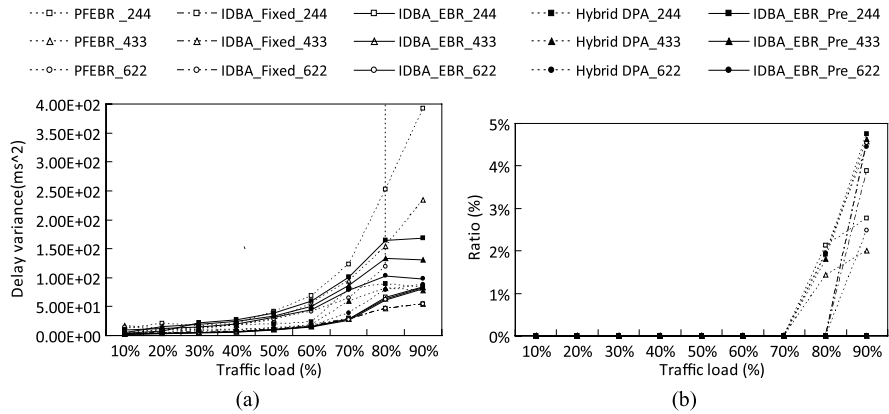
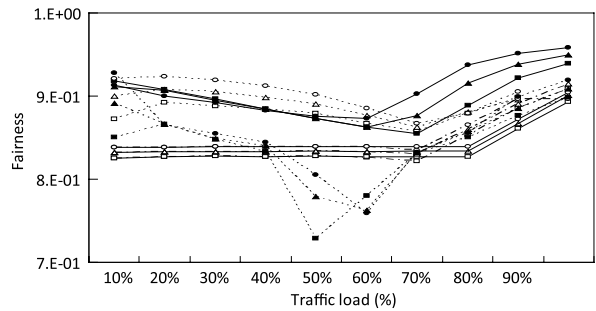


Fig. 8 (a) EF jitter and (b) average packet loss ratio vs. different traffic loads for PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA

Fig. 9 Fairness using Jain’s index vs. different traffic loads for PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA



exceeds 80% in every scenario because lake an effective excess bandwidth reallocation mechanism. Furthermore, IDBA integrates the EBR and remaining bandwidth compensation mechanism to improve bandwidth utilization that prevents packet loss build up in high traffic load for each scenario.

3.4 Fairness

Figure 9 shows the comparison of fairness against different traffic loads among PFEBR, IDBA_Fixed, IDBA_EBR, IDBA_EBR_Pre and hybrid DPA, respectively. Recently, fairness and QoS on DBA schemes have become important issues, which we are also evaluating. The fairness index f ($0 \leq f \leq 1$) has been addressed [22] which is defined as Eq. 8

$$f = \left(\sum_{i=1}^N G_{[i]} \right)^2 / N \sum_{i=1}^N G_{[i]}^2, \tag{8}$$

where N is the total number of ONUs and $G_{[i]}$ is the granted bandwidth of ONU_i . Jain's fairness index f , ranging from 0 to 1, becomes 1 when all ONUs have the same amount of bandwidth allocated by the OLT. Simulation results show that Jain's fairness index f of IDBA is better than hybrid DPA. IDBA_EBR_Pre has the best fairness performance, where the average Jain's fairness index f is about 0.9. One possible reason is that the proposed IDBA_EBR_Pre utilizes the idle period and remaining bandwidth by performing DBA computation for fair bandwidth allocation. The fairness of hybrid DPA begins to gradually vary from traffic load 40% to traffic load 70%. Two possible causes of this is that 1) the hybrid DPA changes the transmission mechanism between online polling and double phase polling and 2) the EBR based on the requested bandwidth in hybrid DPA has some drawbacks, namely unfairness and excessive bandwidth allocated to ONUs over what has been requested.

4 Conclusions

In this study, important factors that can improve the performance of EPON are discussed and evaluated. The IDBA mechanism executes an interleaved transmission process to automatically adjust cycle time to resolve the *idle period problem* for traditional DBA scheme, enhancing the system performance to reduce end-to-end packet delay and improving the throughput. Moreover, it not only accounts for the prediction for differential traffic characteristic but also allocates bandwidth for differential traffic adaptively and improves the utilization of bandwidth by using EBR and the remaining bandwidth compensation mechanism. The simulation results obtained show that the throughput of IDBA is better than that of PFEBR and hybrid DPA, especially in relation to EF traffic performance and average packet loss ratio. Furthermore, the proposed QoS-based prediction mechanism outperforms the hybrid DPA by lowering the EF packet delay. Finally, the IDBA can effectively improve the EF jitter problems faced by PFEBR. It makes use of excess bandwidth and remaining bandwidth to have higher system throughput, lower packet loss and end-to-end delay of each ONU. Future work includes finding out the optimal number of subgroups and having a more appropriate prediction mechanism to improve system performance.

References

1. Green, P.E.: Fiber to the home: The next big broadband thing. *IEEE Commun. Mag.* **42**, 100–106 (2004)
2. Kramer, G., Mukherjee, B., Pessavento, G.: Ethernet PON (ePON): design and analysis of an optical access network. *Photonic Network Commun.* **3**(3), 307–319 (2001)
3. IEEE Draft P802.3ah/D1.0TM: Media Access Control Parameters. Physical Layers and Management Parameters for Subscriber Access Networks, 2002

4. Kramer, G., Mukherjee, B., Pesavento, G.: Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth distribution scheme in an optical access network. *Photonic Network Commun.* **4**(1), 89–107 (2002)
5. Hwang, I.S., Shyu, Z.D., Ke, L.Y., Chang, C.C.: A novel early DBA mechanism with prediction-based fair excessive bandwidth allocation scheme in EPON. *Comput. Commun.* **31**(9), 1814–1823 (2008)
6. Assi, C.M., Ye, Y., Dixit, S., Ali, M.A.: Dynamic bandwidth allocation for quality-of-service over ethernet PONs. *IEEE J. Sel. Areas Commun.* **21**(9), 1467–1477 (2003)
7. Kramer, G., Mukherjee, B., Dixit, S., Ye, Y., Hirth, R.: Supporting differentiated classes of service in ethernet passive optical networks. *J. Opt. Networks* **1**(8), 280–298 (2002)
8. Choi, S.Y., Lee, S., Lee, T.J., Chung, M.Y., Choo, H.: Double-phase polling algorithm based on partitioned ONU subgroups for high utilization in EPONs. *IEEE/OSA J. Opt. Commun. Netw.* **1**(5), 484–497 (2009)
9. Sue, C.C., Cheng, H.W.: A fitting report position scheme for the gated IPACT dynamic bandwidth algorithm in EPONs. *IEEE/ACM Trans. Netw.* **2**(18), 624–637 (2010)
10. McGarry, M., Maier, M., Reisslein, M.: Ethernet PONs: a survey of dynamic bandwidth allocation (DBA) algorithms. *IEEE Commun. Mag.* **42**(8), S8–S15 (2004)
11. Zheng, J.: Efficient bandwidth allocation algorithm for ethernet passive optical networks. *IEE Proc. Commun.* **153**(3), 464–468 (2006)
12. Luo, Y., Ansari, N.: Bandwidth allocation for multiservice access on EPON. *IEEE Commun. Mag.* **43**(2), S16–S21 (2005)
13. Hwang, J., Yoo, M.: QoS-aware class gated DBA algorithm for the EPON system. In: *International Conference on Advanced Technologies for Communications*, pp. 363–366 (2008)
14. Chen, J., Chen, B., Wosinska, L.: Joint bandwidth scheduling to support differentiated services and multiple service providers in 1G and 10G EPONs. *J. Opt. Commun. Netw.* **1**(4), 343–351 (2009)
15. Kramer, G.: *Ethernet Passive Optical Networks*. McGraw-Hill Professional, New York, ISBN:0071445625 (2005)
16. Naser, H., Moustah, H.T.: A joint-ONU interval-based dynamic scheduling algorithm for ethernet passive optical networks. *IEEE/ACM Trans. Netw.* **14**(4), 889–899 (2006)
17. Ma, M., Zhu, Y., Cheng, T.H.: A bandwidth guaranteed polling MAC protocol for ethernet passive optical networks. In: *Proc. IEEE INFOCOM*, San Francisco, CA, pp. 22–31 (2003)
18. ITU-T Recommendation G.114: One-way transmission time. In: *Series G: Transmission Systems and Media, Digital Systems and Networks*. Telecommunication Standardization Sector of ITU (2000)
19. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for differentiated services, RFC 2475. www.ietf.org/rfc/rfc2475.txt (1998)
20. Willinger, W., Taqqu, M.S., Erramilli, A.: A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks. In: *Stochastic Networks: Theory and Applications*. R. Statist. Soc. Lect. Notes Ser., vol. 4. Oxford University Press, London (1996)
21. Bai, X., Shami, A.: Modeling self-similar traffic for network simulation. Technical report, NetRep-2005-01 (2005)
22. Chen, B., Chen, J., He, S.: Efficient and fine scheduling algorithm for bandwidth allocation in Ethernet passive optical networks. *IEEE J. Sel. Topics Quant. Electron.* **12**(4), 653–660 (2006)
23. Jain, R., Durrresi, A., Babic, G.: Throughput fairness index: an explanation. http://www.cs.wustl.edu/jain/atmf/ftp/af_f

The Game of n -Player Shove and Its Complexity

Alessandro Cincotti

Abstract Why are n -player games much more complex than two-player games? Is it much more difficult to cooperate or to compete? The game of n -player Shove is the n -player version of Shove, a two-player combinatorial game. In multi-player games, because of the possibility to form alliances, cooperation between players is a key-factor to determine the winning coalition and, as a consequence, n -player Shove played on a set of finite strips is \mathcal{PSPACE} -complete.

Keywords Combinatorial games · Complexity · n -Player Shove

1 Introduction

The game of Shove is a combinatorial game defined in [1] and played on a finite strip of squares where each square is empty or occupied by a red or blue piece. Two players, called Red and Blue, move alternately. A piece of the current player's color and all pieces to its left are pushed one square to the left. Pieces may be shoved off the end of the strip. The first player unable to move is the loser. An example of Shove is shown in Fig. 1.

n -Player Shove is the n -player version of Shove played on a set of finite strips of squares where each square is empty or occupied by a piece labeled by an integer $j \in \{1, 2, \dots, n\}$.

The first player pushes a piece labeled 1 and all pieces to its left one square to the left. Pieces may be shoved off the end of the strip. The second player pushes a piece labeled 2 and all pieces to its left one square to the left. The other players move in similar way.

A. Cincotti (✉)

School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
e-mail: cincotti@jaist.ac.jp

Fig. 1 An example of Shove

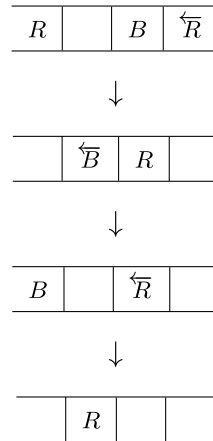


Fig. 2 An example of queer game



Players take turns making legal moves in cyclic fashion (1-st, 2-nd, ..., n -th, 1-st, 2-nd, ...). When one of the players has no legal play, that player leaves the game and the remaining $n - 1$ players continue playing in the same mutual order as before. The remaining player is the winner.

We briefly recall the definition of *queer* game introduced by Propp [16]:

Definition 1 A position in a three-player combinatorial game is called queer if no player can force a win.

Such a definition is easily generalizable to n players:

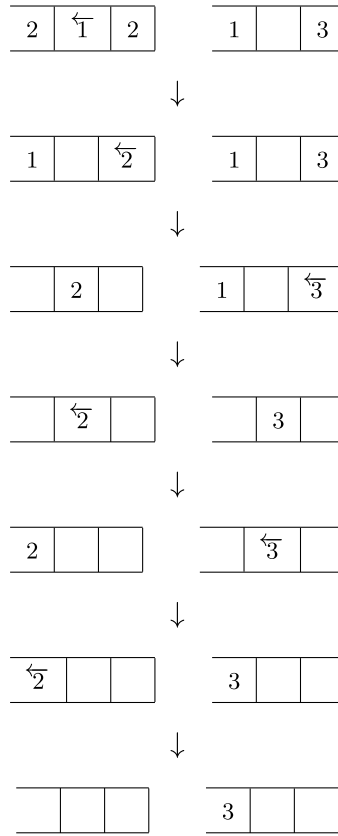
Definition 2 A position in an n -player combinatorial game is called queer if no player can force a win.

In the game of n -player Shove is not always possible to determine the winner because of queer games. In the game shown in Fig. 2 two different scenario are possible:

- If the first player move in the first strip, then the third player wins as shown in Fig. 3.
- If the first player moves in the second strip, then the second player wins as shown in Fig. 4.

In two player games [2, 11] players are in conflict to each other and coalitions are not allowed but in n -player games [3, 13, 14, 17], when the game is queer, only cooperation between players can guarantee a winning strategy, i.e., one player of the coalition is always able to make the last move. As a consequence, to establish whether or not a coalition has a winning strategy is a crucial point.

Fig. 3 The third player wins



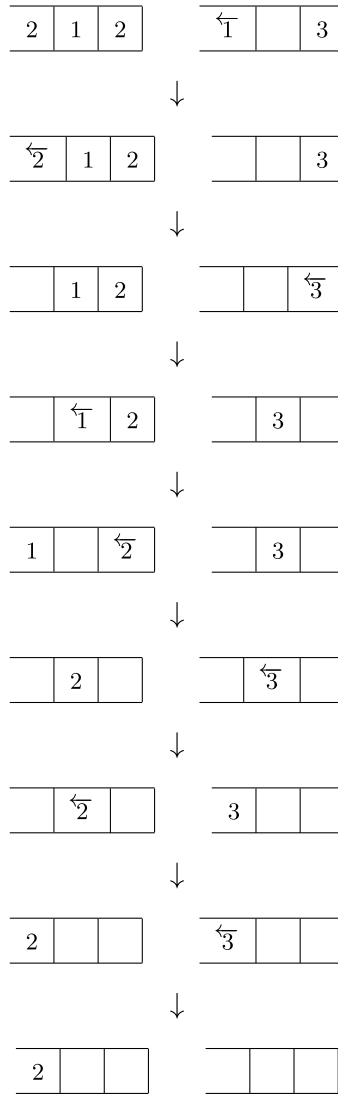
In previous works, we analyzed the complexity of n -player Hackenbush played on strings [4, 5], n -player map-coloring games [6–8], and n -player Toppling Dominoes played on rows [9, 10]. In this paper we show that, in Shove, cooperation can be much more difficult than competition and, as a consequence, n -player Shove is \mathcal{PSPACE} -complete.

2 The Complexity of n -Player Shove

In this section we show that the \mathcal{PSPACE} -complete problem of *Quantified Boolean Formulas* [12, 15], QBF for short, can be reduced by a polynomial time reduction to n -player Shove.

Let $\varphi \equiv \exists x_1 \forall x_2 \exists x_3 \dots Q x_n \psi$ be an instance of QBF, where Q is \exists for n odd and \forall otherwise, and ψ is a quantifier-free Boolean formula in conjunctive normal form where every clause has 3 distinct literals. We recall that QBF asks if there exists an assignment to the variables $x_1, x_3, \dots, x_{2\lceil n/2 \rceil - 1}$ such that the formula evaluates to true. If n is the number of variables and k is the number of clauses in ψ , then the

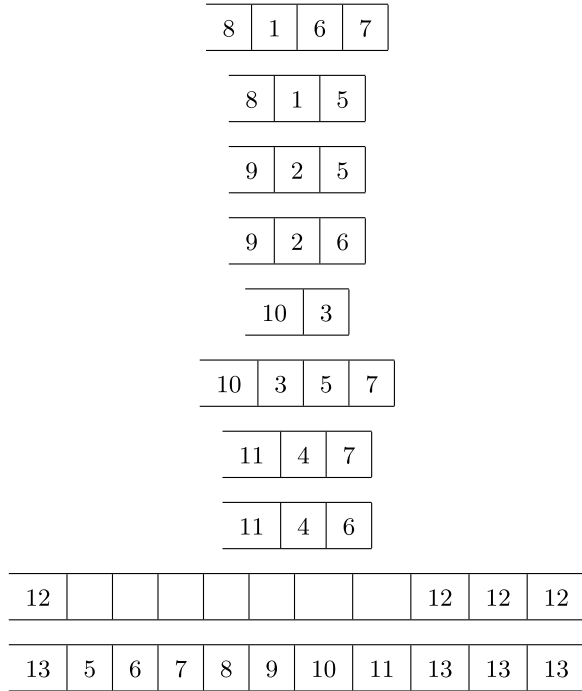
Fig. 4 The second player wins



instance of n -player Shove will have $2n + k + 2$ players and $2n + 2$ strips, organized as follows:

- For each variable x_i , with $1 \leq i \leq n$, we add two new strips. Both strips have a piece labeled $n + k + i$, with $n + k + 1 \leq n + k + i \leq 2n + k$ in the first square on the left and a piece labeled i , with $1 \leq i \leq n$ in the second square on the left. In the first strip we add one piece for each clause that contains x_i and in the second strip we add one piece for each clause that contains \bar{x}_i . These pieces are labeled j , with $n + 1 \leq j \leq n + k$, and arranged in increasing order from left to right.

Fig. 5 An example of reduction



- The $2n + 1$ -th strip contains 4 pieces labeled $2n + k + 1$ respectively in the first, $(n + k + 2)$ -th, $(n + k + 3)$ -th, and $(n + k + 4)$ -th square of the strip.
- The last strip contains $n + k + 4$ pieces labeled from left to right $2n + k + 2, n + 1, \dots, 2n, 2n + 1, \dots, 2n + k, 2n + k + 2, 2n + k + 2, 2n + k + 2, 2n + k + 2$.

Let us suppose that:

- The first coalition is formed by $\lfloor n/2 \rfloor + 1$ players corresponding to the 2-nd, 4-th, $\dots, 2\lfloor n/2 \rfloor$ -th, and $2n + k + 1$ -th players,
- The second coalition is formed by the remaining players.

An example of reduction is shown in Fig. 5 where

$$\varphi \equiv \exists x_1 \forall x_2 \exists x_3 \forall x_4 (C_5 \wedge C_6 \wedge C_7)$$

and

$$C_5 \equiv (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

$$C_6 \equiv (x_1 \vee \overline{x_2} \vee \overline{x_4})$$

$$C_7 \equiv (x_1 \vee \overline{x_3} \vee x_4)$$

The problem to determine the winning coalition is strictly connected to the problem of QBF, as shown in the following theorem.

Theorem 1 *Let G be a general instance of n -player Shove played on a set of strips. Then, to establish whether or not a given coalition has a winning strategy is a \mathcal{PSPACE} -complete problem.*

Proof We show that it is possible to reduce every instance of QBF to a graph G representing an instance of n -player Shove. Previously we have described how to construct the instance of n -player Shove, therefore we just have to prove that QBF is satisfiable if and only if the second coalition has a winning strategy.

If QBF is satisfiable, then there exists an assignment of x_i such that ψ is true with $i \in \{1, 3, \dots, 2\lceil n/2 \rceil - 1\}$. If x_i is true, then the i -th player plays in the strip with the pieces corresponding to the clauses containing x_i . If x_i is false, then the i -th player plays in the strip with the pieces corresponding to the clauses containing \bar{x}_i . Every clause contains at least a true literal, therefore the i -th player with $i \in \{n+1, n+2, \dots, n+k\}$ can always push a piece in the strip corresponding to that literal. In this way, at the end of the first round, the $2n+k+2$ -th player is able to shove the piece in the first square of the last strip and, at the end of the game, he/she will be able to make the last move. Therefore, the second coalition has a winning strategy.

Conversely, let us suppose that the second coalition has a winning strategy. We observe that the $2n+k+1$ -th player is always able to make $3(n+k)+7$ moves, therefore the $2n+k+2$ -th player must be able to make the same number of moves in order to guarantee a winning strategy for the second coalition. As a consequence, the i -th player with $i \in \{n+1, \dots, 2n+k\}$ does not shove any piece in the last strip during the first round, i.e., every clause has at least one true literal and QBF is satisfiable.

Therefore, to establish whether or not a coalition has a winning strategy in n -player Shove played on a set of strips is \mathcal{PSPACE} -hard.

To show that the problem is in \mathcal{PSPACE} we present a polynomial-space recursive algorithm to determine which coalition has a winning strategy as shown in Fig. 6.

Let us introduce some useful notations:

- $G = (V, E)$ is the graph representing an instance of n -player Shove;
- p_i is the i -th player;
- C_0 is the set of current players belonging to the first coalition;
- C_1 is the set of current players belonging to the second coalition;
- $\text{coalition}(p_i)$ returns 0 if $p_i \in C_0$ and 1 if $p_i \in C_1$;
- $\text{label}(v)$ returns the label of the piece v ;
- $\text{next}(p_i)$ returns the player which has to play after p_i ;
- $\text{shove}(G, v)$ returns the graph obtained from G after that the piece v has been shoved.

Algorithm `Check` receives in input a graph $G = (V, E)$, the two initial coalitions C_0 and C_1 , and the player p_i that has to move. It returns 0 if the first coalition wins and 1 if the second coalition wins.

Fig. 6 A polynomial-space recursive algorithm

```

Algorithm Check( $G, C_0, C_1, p_i$ )
   $j \leftarrow \text{coalition}(p_i)$ ;
  if  $\nexists v \in V : \text{label}(v) = i$  then
     $C_j \leftarrow C_j \setminus \{p_i\}$ ;
    if  $C_j = \emptyset$  then
      return  $1 - j$ ;
    else
      return Check( $G, C_0, C_1, \text{next}(p_i)$ );
    end
  else
    for all  $v \in V : \text{label}(v) = i$  do
       $G' \leftarrow \text{shove}(G, v)$ ;
      if Check( $G', C_0, C_1, \text{next}(p_i)$ ) =  $j$  then
        return  $j$ ;
      end
    end
    return  $1 - j$ ;
  end
    
```

Algorithm Check performs an exhaustive search until a winning strategy is found and its correctness can be easily proved by induction on the depth of the game tree.

Algorithm Check is clearly in \mathcal{PSPACE} because the number of nested recursive calls is at most $|V|$ and therefore the total space complexity is $O(|V|^2)$. \square

References

1. Albert, M.H., Nowakowski, R.J., Wolfe, D.: Lessons in Play: An Introduction to Combinatorial Game Theory. AK Peters, Wellesley (2007)
2. Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning Way for Your Mathematical Plays. AK Peters, Wellesley (2001)
3. Cincotti, A.: Three-player partizan games. Theoret. Comput. Sci. **332**(1–3), 367–389 (2005)
4. Cincotti, A.: Three-player Hackenbush played on strings is \mathcal{NP} -complete. In: Ao, S.I., Castillo, O., Douglas, C., Feng, D.D., Lee, J. (eds.) Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, IMECS 2008, Hong Kong, 19–21 March 2008, pp. 226–230 (2008). Newswood Limited
5. Cincotti, A.: On the complexity of n -player Hackenbush. Integers **9**, 621–627 (2009)
6. Cincotti, A.: On the complexity of three-player snort played on complete graphs. In: Li, W., Zhou, J. (eds.) Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, 8–11 August 2009, pp. 68–70. IEEE Press, New York (2009).
7. Cincotti, A.: Three-player col played on trees is \mathcal{NP} -complete. In: Ao, S.I., Castillo, O., Douglas, C., Feng, D.D., Lee, J. (eds.) Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, IMECS 2009, Hong Kong, 18–20 March 2009, pp. 445–447 (2009). Newswood Limited

8. Cincotti, A.: On the complexity of some map-coloring multi-player games. In: Ao, S.I., Castillo, O., Huang, H. (eds.) *Intelligent Automation and Computer Engineering*. LNEE, vol. 52, Springer, Berlin (2010)
9. Cincotti, A.: On the complexity of n -player toppling dominoes. In: Ao, S.I., Castillo, O., Douglas, C., Feng, D.D., Lee, J. (eds.) *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 461–464 (2010). Newswood Limited
10. Cincotti, A.: Three-player toppling dominoes is \mathcal{NP} -complete. In: Mahadevan, V., Zhou, J. (eds.) *Proceedings of the 2nd International Conference on Computer Engineering and Technology*, Chengdu, 16–18 April 2010, pp. 548–550. IEEE Press, New York (2010)
11. Conway, J.H.: *On Numbers and Games*. AK Peters, Wellesley (2001)
12. Garey, M.R., Johnson, D.S.: *Computers and Intractability*. Freeman, New York (1979)
13. Li, S.Y.R.: n -Person nim and n -person moore's games. *Int. J. Game Theory* **7**(1), 31–36 (1978)
14. Loeb, D.E.: Stable winning coalitions. In: Nowakowski, R.J. (ed.) *Games of No Chance*, pp. 451–471. Cambridge University Press, Cambridge (1996)
15. Papadimitriou, C.H.: *Computational Complexity*. Addison-Wesley, Reading (1994)
16. Propp, J.G.: Three-player impartial games. *Theoret. Comput. Sci.* **233**(1–2), 263–278 (2000)
17. Straffin Jr., P.D.: Three-person winner-take-all games with McCarthy's revenge rule. *Coll. Math. J.* **16**(5), 386–394 (1985)

Modeling the Vestibular Nucleus

Alexandru Codrean, Adrian Korodi,
Toma-Leonida Dragomir, and Vlad Ceregan

Abstract In recent years the vestibular-sympathetic reflex has received an increasing amount of attention due to the role it could play in the human organism in different types of scenarios. Despite this, quantitative models of this reflex mechanism are still lacking. In this context, the current paper aims at taking a first step towards the modeling of the vestibular-sympathetic reflex by developing a model of the Vestibular Nucleus – the central part of the vestibular-sympathetic reflex. After a careful analysis of the limitations and uncertainties of the available experimental data from the literature, a three step modeling methodology for the Vestibular Nucleus is presented. After a description of the operations involved in each step, in the end, some preliminary results are shown.

Keywords Vestibular nucleus · Interpolation · Firing rate · Vestibular-sympathetic reflex

1 Introduction

With the integration between neuroscience, engineering and medicine, new discoveries are emerging at a rapid research gradient, that evolve from fundamental knowledge of the role and behavior of different neural structures, to modeling aspects towards a quantitative representation, and finally to applications that could be further used in clinical practice. In this direction, many studies are focusing on how the information is processed in different nervous centers of the human brain, and what kinds of signals carry this information to different organs (e.g. the heart). Usually, the physiological mechanisms through which these nervous control signals influence the end-effector organs are called nervous reflex mechanisms, or in some cases nervous regulatory mechanisms. Thus, a nervous reflex mechanism is

A. Codrean (✉)
“Politehnica” University of Timisoara, 2 Vasile Parvan Bld, 300223 Timisoara, Romania
e-mail: alexandrucodrean@yahoo.com

associated with a specific nervous center (or more) and with the efferent organs involved.

An important nervous center involved in several reflex mechanisms like the vestibular-spinal reflex, vestibular-ocular reflex, and the vestibular – sympathetic reflex is the Vestibular Nucleus (part of the Vestibular System-VS). Recent studies have focused on the role of the vestibular-sympathetic reflex, and its effect on the cardiovascular system (more specifically the heart) [1, 2]. However, assessing the influence of the vestibular-sympathetic reflex on the cardiovascular system (CVS) poses difficulties due to its interaction with the baroreflex, which is one of the main nervous regulatory mechanisms of the CVS on short-term.

From these two nervous reflex mechanisms, only the baroreflex is understood at a quantitative level, and several models exist in the literature that have been validated with experimental data (e.g. [3]). For the vestibular-sympathetic reflex no current model exists in the literature.

In order to obtain a more complete model for the nervous control of the CVS that could then be used in different clinical scenarios (e.g. orthostatic stress-head up tilt test/sitting to standing), a quantitative model of the vestibular sympathetic-reflex must be developed, that would be integrated with an existing baroreflex model.

In this context, this paper aims in taking a first step in the direction of a quantitative modeling of the vestibular-sympathetic reflex, by focusing on developing a model of the Vestibular Nucleus (the central part of the VS and the most important level of information processing of the vestibular-sympathetic reflex).

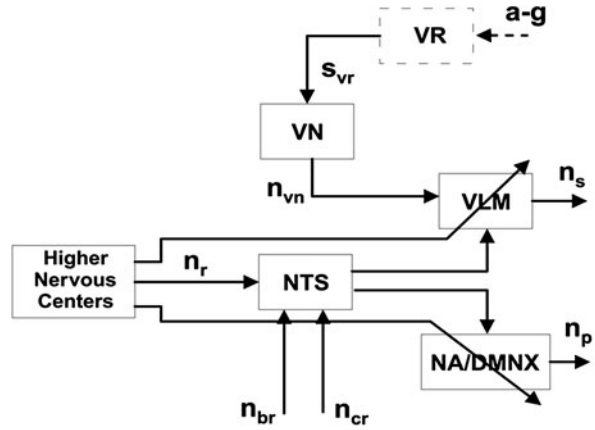
2 Information Processing Through the Vestibular-Sympathetic Reflex

2.1 The Main Components of the Vestibular-Sympathetic Reflex

The interactions between the vestibular-sympathetic reflex mechanism and the baroreflex mechanism occur at the level of the Medulla Oblongata (MO) and, from an informational point of view, they are illustrated in the block diagram from Fig. 1.

The nervous centers in the MO playing a role in cardiovascular regulation are: the Nucleus Tractus Solitarius (NTS), the Nucleus Ambiguus/Dorsal Motor Nucleus of the vagus (NA/DMNX), the Ventrolateral Medulla (VLM) and the Vestibular Nucleus (VN). The first level of information processing for the baroreflex is the NTS, while for the vestibular-sympathetic reflex, it's the VN. The NTS receives information from the baroreceptors (n_{br} signal) and from the cardiopulmonary receptors (n_{cp} signal), whereas the VN receives information from the vestibular receptors VR – (s_{vr} vector signal). The VR in turn detect a change in head acceleration – a signal. The second level of information processing for these two nervous reflex mechanisms is the VLM, and their effects are believed to be additive [1]. The output signal of

Fig. 1 Nervous centers in the MO involved in the nervous control of the CVS



the VLM is transmitted to the Sympathetic System, while the output signal of the NA/DMNX is transmitted to the Parasympathetic System. Afterwards, the Sympathetic and the Parasympathetic Systems transmit the nervous control signals to the CVS.

The reference signal n_r , coming from Higher Nervous Centers (HNC), provides a variable set-point for blood pressure. As the figure suggests, the HNC may influence the behavior of the VLM and the NA/DMNX, according to certain conditions (e.g. psychological factors).

Finally, it can be concluded that the vestibular-sympathetic reflex is composed out of 4 main parts: VR, VN, VLM and the Sympathetic System. Regarding the VLM, not much is currently known, except that it's a possible convergence level between the baroreflex and vestibular-sympathetic reflex and, as we mentioned earlier, that this convergence could be considered in a simplified manner as being additive. For the VR quite a few models are presented in the literature [4–6].

The VN has not yet been modeled in a quantitative manner, despite the fact that it could be regarded as the most important level of information processing for the vestibular-sympathetic reflex. Although it is considered to have many neurological functions, with strong interconnections with other nervous centers, only its contribution to the vestibular-sympathetic reflex is of interest for the current presentation. In this context, the most important step in the direction of a quantitative modeling of the vestibular-sympathetic reflex is to firstly determine a model for the VN.

Before we start with the modeling aspects, and how we used the experimental results from [7] for developing a model for the VN, it's important to analyze what types of uncertainties can arise from neurophysiologic experimental data presented in the literature (e.g. [7]).

2.2 *Assessing the Uncertainties and Limitations in Neurophysiologic Experimental Data*

A certified fact is that many systems were designed guided by an expert knowledge. Often, there is no other way to describe various functioning patterns and they are best expressed through information taken over from experiments. Therefore, mathematical models of different nature can be obtained using the data provided by an expert. Generally, the steps that have to be followed in order to design a mathematical model in the domain of biological systems, particularly the field of neurophysiology, are strictly dependent on the information provided by an expert. The expert provides experimental data which can be processed and used to develop the model.

Let's now consider the black box representation of a Neural System from Fig. 2a. The input signal is a current, which usually has a slow variation in time (with the exception when step input signals are used) in the range of nA. The output signal is a voltage, and it can either remain at a relative constant negative value (e.g. -70 mV), or voltage spikes can occur (called action potentials AP). When the output signal is at the constant negative level, the Neural System is considered to be inhibited, while when the output signal contains voltage spikes the System is excited. Thus, the main interest is to study the Neural System when output spikes or action potentials occur.

It is well known that neurons generate voltage spikes at constant amplitude, and that the information received from the input current signal is practically modulated into the instantaneous spike rate of the output voltage spikes [8]. If the shape of the action potential is important from a cellular (micro) point of view in understanding the biochemical mechanisms involved, from a physiological (macro) point of view only the instantaneous rate is needed in order to describe the information processing that characterizes the Neural System. Figure 2b shows an alternative black box representation of the Neural System, where the output signal is a computed instantaneous rate of the voltage spikes (also called instantaneous firing rate – IFR).

In neurophysiology, experimental data obtained for individual neurons or a small group of neurons (which can be embedded in the above mentioned Neural System) is usually presented into two ways:

- in the **time domain** – when the input current signal is usually a step or a ramp signal, and the output signal used is either voltage or the IFR;
- in the **frequency domain** – when the input current signal is of sinusoidal form and the output signal used – the firing rate – is also of the same form, experimental frequency characteristics can be obtained.

The shortcoming of time domain experiments is that the output signal used in most cases is voltage, which is not convenient for physiological modeling for the above mentioned reasons, and experiments in which the firing rate was used as output signal are very rare and with a limited number of scenarios (e.g. ramp input current for 3 slope values). So developing a physiological model based on time domain experimental data is rather difficult, unless one finds experiments for the Neural System using the firing rate as output in the literature.

The frequency domain experiments pose the advantage of using the firing rate as output signal, while frequency characteristics bring more insight into the overall dynamics of the system, but other types of drawbacks arise here. Firstly, the hypothesis that the Neural System is linear in the vicinity of the current operating regime has to be adopted. Only then frequency domain identification methods could be applied in developing a model. Secondly, due to the technological limitations of the measurement equipment, the experiments can be done on a limited frequency range (e.g. the frequency range for the experiments done on the VN neurons is somewhere between 0.1 Hz and 50 Hz). And thirdly, the number of operating points for which such experiments are made is limited. If we consider the mean firing rate as an index of the system's operating point, then for the VN neurons the mean firing rate range is usually between 10 spikes/sec and 100 spikes/sec (having in mind that the Neural System is capable of reaching mean firing rates of greater values than 100 spikes/sec). This could all influence the model identification process at a certain degree varying from case to case.

Despite the drawbacks of the frequency domain experiments, at present, this type of experimental data seems to provide the most quantitative information for our case study – the Vestibular Nucleus. Thus, we will further focus on using frequency domain experiments as a starting point in developing a model for the VN, and the time domain experiments will only be used later one for additional model validation.

Now, that we have discussed about the limitations of the neurophysiologic experimental data found in the literature, we will move on to analyzing the uncertainties that may also appear.

Uncertainty, from the control theory point of view, can be divided in two main categories: process uncertainty and signal uncertainty [9]. Process uncertainty relates to how the physical system (or process) handles the information received through the input signals and passes it to the output (a model is always a simplified representation of the physical system). Signal uncertainty refers to “external” influences such as unmeasured disturbances or noise corrupting measurements and sometimes it can be even introduced involuntary through the adopted measurement technique (or experimental set-up for data acquisition).

Let's now return to the Neural System from Fig. 2b, in the case when we want to use experimental frequency characteristics for identifying a model. We mentioned that the IFR output signal is not a physical signal, because it is actually computed based on the real physical output Voltage signal (Fig. 2a). The manner in which this calculus (or association) is done is not treated unanimously in the literature. Practically, the IFR signal consists of discrete points that correspond to the interval between two consecutive Voltage Spikes, and these points are eventually interpolated in order to obtain a continuous time signal. The main issue here is to what time moment these discrete points should be associated.

For example, for our VN system, we found two main approaches for associating these time moments (shown in Fig. 3a and b) in the literature. Let's consider that for a train of five voltage spikes (corresponding to five time moments t_1, \dots, t_5) the IFR must be computed based on the time intervals between two consecutive voltage spikes ($T_1 = 40$ ms, $T_2 = 50$ ms, $T_3 = T_4 = 60$ ms). Because we have four time

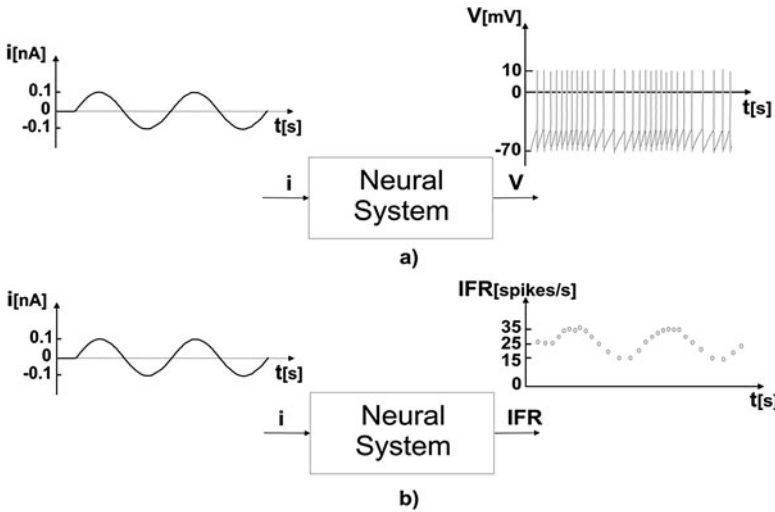


Fig. 2 Black box representation of a Neural System

intervals, we will obtain four discrete values for the IFR signal: $\text{IFR}_1 = 1/T_1 = 25$ spikes/s, $\text{IFR}_2 = 1/T_2 = 20$ spikes/s, $\text{IFR}_3 = 1/T_3 = 16.7$ spikes/s and $\text{IFR}_4 = 1/T_4 = 16.7$ spikes/s.

Figure 3a shows the first approach used in [7] by du Lac et al., when each discrete point of the IFR is associated to the time moment of the spike at the beginning of each interval. So the four discrete values IFR_1 – IFR_4 correspond to the time moments t_1 – t_4 .

Figure 3b shows the second approach used in [10] by Uno et al. and in [11] by Ris et al., when each discrete point of the IFR is associated to the time moment of the spike at the end of each interval. So the four discrete values IFR_1 – IFR_4 correspond to the time moments t_2 – t_5 .

Now if we compare the two methods, we can see in Fig. 3c that the first method introduces a lead component (anticipative behavior), while the second method introduces a lag component (a latency). Thus, the method in which the IFR signal is computed introduces a signal uncertainty. In order to take into account this uncertainty we must define an uncertainty area – in this example, this area is marked by curves (1) and (2), which refer to the two mentioned methods. The true evolution of the IFR might be somewhere in the middle (the curve with solid line), where the time values associated to the IFR values would be chosen at the middle of the T intervals.

The errors introduced by this type of uncertainty can vary depending on the spectrum of the input signal. This can be illustrated with the help of frequency characteristics, when the input signal and the output signal have the sinusoidal form from Fig. 2b. If we use the experimental frequency characteristics for the VN obtained in [7], we can calculate with approximation the frequency characteristics that would have been obtained by using the second method of computing the IFR signal. This

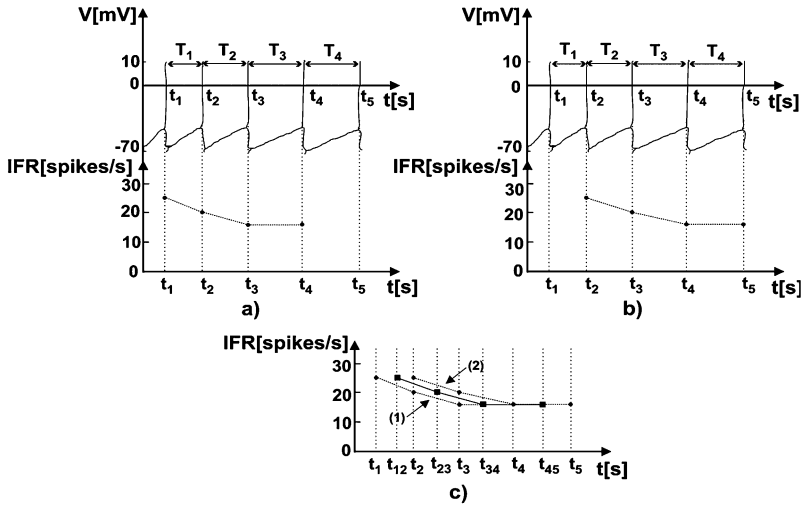


Fig. 3 Computing the IFR signal ((1) after [7], (2) after [10])

calculus is applied only to the phase-frequency characteristics, and is based on the following formula:

$$\begin{aligned} \varphi_2(\omega_i) &= \varphi_1(\omega_i) + \Delta\varphi(\omega_i), \quad \omega_i \in [0.2\pi, 20\pi] \text{ rad/s} \\ \Delta\varphi(\omega_i) &= -\omega_i \cdot \Delta T_m \cdot \frac{180}{\pi} \end{aligned} \tag{1}$$

where ω_i represents a frequency value, φ_1 and φ_2 are the phase characteristics with the first and second method, $\Delta\varphi$ is a phase adjustment and ΔT_m is the mean interval between two consecutive voltage spikes for a given output signal variation (which in this case is equal to the reciprocal of the mean firing rate – MFR).

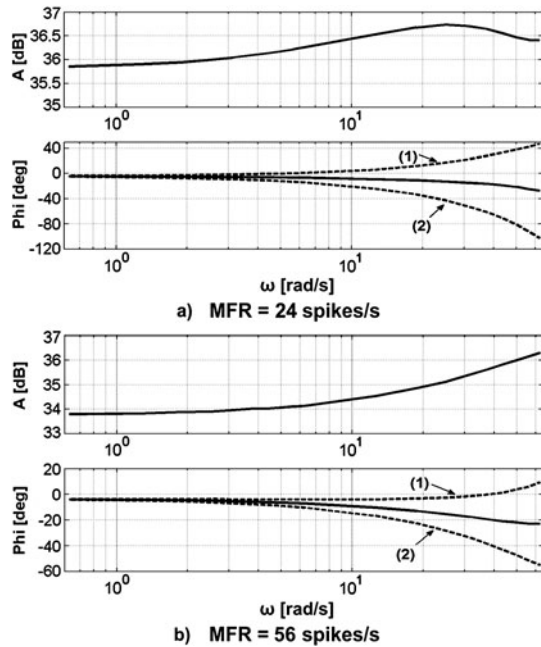
Figure 4 shows the experimental frequency characteristics for two MFRs obtained in [7] with method 1 (Amplitude-frequency characteristic – solid line, and Phase-frequency characteristic – dashed line), the Phase-frequency characteristics calculated using (1) that would correspond to the second method (also dashed line), and the Phase-characteristics that would be obtained if the middle of the inter-spike interval had been associated to the IFR discrete values (solid line). These last Phase-frequency characteristics have been obtained from the previous two with the formula:

$$\varphi_c(\omega_i) = \varphi_2(\omega_i) + 0.5 \cdot (\varphi_1(\omega_i) - \varphi_2(\omega_i)), \quad \omega_i \in [0.2\pi, 20\pi] \text{ rad/s} \tag{2}$$

Here, the uncertainty zone is marked by the Phase-frequency characteristics obtained with methods one and two (dashed lines). It can be stated that the errors due to signal uncertainties greatly increase with frequency. These results are in agreement with recent studies [12].

All in all, if we want to derive a mathematical model (for the VN in particular) from experimental data in the frequency domain, we must take into account the

Fig. 4 Experimental frequency characteristics for the VN neurons with uncertainty areas



uncertainty introduced by the method of calculating the IFR output signal. One way of doing this is by correcting the Phase-Frequency characteristics with the relations (1) and (2). It's also important to underline each time, the adopted hypothesis.

2.3 A Modeling Methodology for the Vestibular Nucleus

In this section we analyzed, with a focus on experimental frequency characteristics, how experimental data available in the literature can be used to develop a model of the VN. In order to provide a mathematical model for the channel $s_{vr} \rightarrow n_{vn}$ of the VN (see Fig. 1), the experimental data from [7] was considered the most appropriate knowledge that could constitute a starting point for developing a proper model of the VN. For a more rigorous organization of the modeling process, we first established a three step methodology (3S-M) – Fig. 5.

Through this methodology a fixed structure model with variable parameters able to reproduce the adaptive behavior of the VN at the input signal s_{vr} will be obtained. This should be understood as follows: the physical signal s_{vr} can be described using two information carrier variables, the mean firing rate s_{mfr} and the stimulation input current \tilde{s}_{vr} , aggregated as a vector signal $s_{vr} = [s_{mfr} \tilde{s}_{vr}]$ (Fig. 1 and Fig. 6). The output variable of VN, the signal n_{vn} , is obtained by processing \tilde{s}_{vr} using a processing algorithm whose parameters are continuously adapted to the state of the VS represented by s_{mfr} . In this context the model could be regarded as an adaptive one.

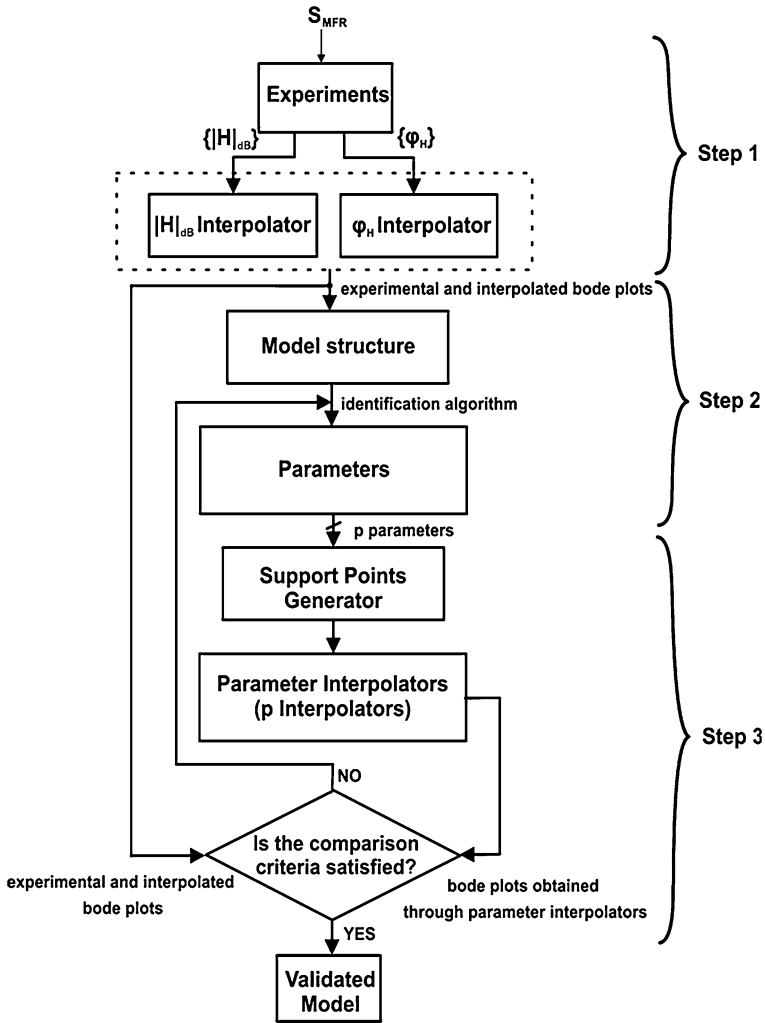


Fig. 5 Methodology for obtaining a complete model for the VN from experimental frequency characteristics of individual neurons

In the first step of the 3S-M, based on the work from [13], a pair of interpolative Bode plots generators will be developed based on the few Bode plots determined experimentally in [7] for different values of the mean firing rate (s_{mfr}). The generators will provide a nonparametric model in the form of amplitude-frequency characteristics ($\{|H|_{dB}\}_c$) and phase-frequency characteristics ($\{\phi_H\}_c$), calculated for different values of s_{mfr} . The ensemble of interpolators will approximate the frequency response of the VN in the range $s_{mfr} \in [10-56 \text{ spikes/s}]$.

In the second step of the 3S-M, based on the work from [14], a continuous time model of the VN will be outlined. The continuous time model will describe the

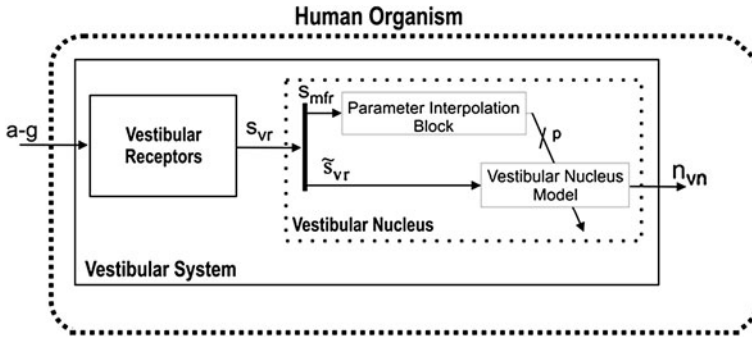


Fig. 6 The integration of the complete VN model in the Human Organism

input-output dependency $\tilde{s}_{vr} \rightarrow n_{vn}$, using p parameters that depend on s_{mfr} . For a given pair of Bode plots (i.e. a given value of s_{mfr}) an identification algorithm will be used to compute the p parameters.

The aim of the third step of 3S-M (the work from [15]) will consist in an extension of the model developed in Step 2 in a manner that will avoid the usage of the identification algorithm in current simulations. To do this, each of the p parameters will be determined as a function of s_{mfr} by its own interpolator (p Parameter Interpolators or p-PI). This would permit the model to adapt the parameters to the current value of s_{mfr} .

3 Developing a Model of the Vestibular Nucleus

3.1 Step One of the 3S-M

As we mentioned in the previous section, we will use the experimental frequency characteristics from [7] as starting point. These frequency characteristics were obtained for a small number of s_{mfr} values, by considering the following: the input signal \tilde{s}_{vr} is of sinusoidal form in respect with the time $t - S_0 \cdot \sin(2\pi n_{vr}t)$ (with S_0 being the signal's amplitude), the output signal n_{vn} is also a sinusoidal signal $n_{vn}(t) = N_{vn0} \cdot \sin(2\pi n_{vr}t + \phi_{vn})$ (N_{vn0} is the signal's amplitude and the phase ϕ_{vn} reflects the inertia that appears in the transmission of \tilde{s}_{vr} through the VN).

In order to obtain various frequency characteristics of the VN activity, two interpolative generators will be used (see Fig. 7). Both interpolative generators, $GIP_{Amplitude}$ and GIP_{Phase} , make use of two-dimensional lookup-tables that are constituted using the experimental data from [7]. The inputs of the interpolative tables are: the mean firing rate s_{mfr} and an instantaneous frequency signal n_{vr} . The outputs are the amplitude N_{vn0} and the phase ϕ_{vn} .

As it can be seen in Fig. 7, the first input of each generator is s_{mfr} , a signal with slow variation in time. Therefore, according to the experiments related in [7], next in the scenario, constant values will be considered for s_{mfr} , particularly: 10,

Fig. 7 Interpolative type generators for frequency characteristics of the VN activity

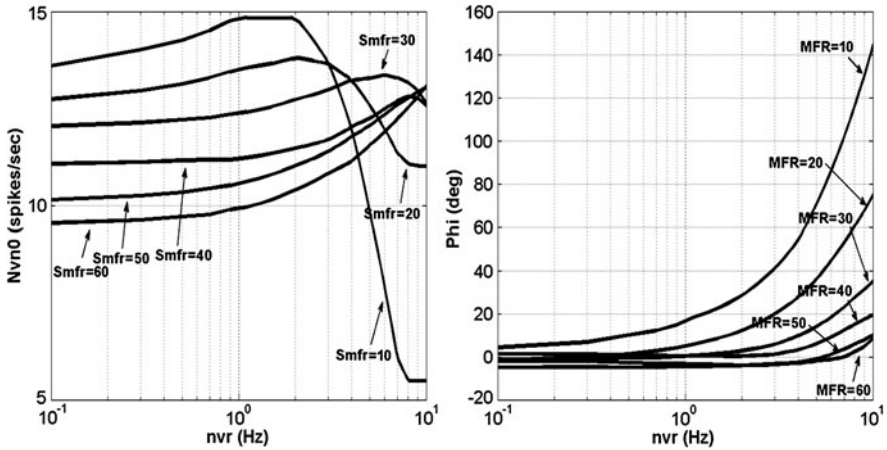
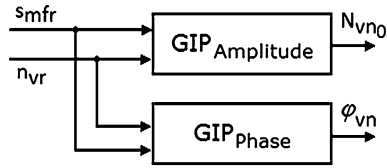


Fig. 8 Frequency characteristics obtained with linear interpolation using GIP

20, 30, 40, 50 and 60 spikes/sec. Figure 8 illustrates the amplitude-frequency and phase-frequency characteristics generated with $GIP_{Amplitude}$ and GIP_{Phase} obtained through linear interpolation.

The ensemble of frequency characteristics highlights the hypothesis that the behavior of the neurons may consist in jumps between characteristics corresponding to different MFRs. Moreover, it seems that the MFR of a neuron or a group of neurons can be correlated to its current operating point.

Now, that we have established the experimental and interpolated frequency characteristics that we want to use in the modeling process, the uncertainties we mentioned in Sect. 2 can be taken into account. Because in [7], du Lac used the first method in computing the IFR signal, before we proceed to step two of the 3S-M, we must obtain the corrected Phase-frequency characteristics (through which the uncertainties are taken into account). This can be easily done, for Phase-frequency characteristics corresponding to any value of the s_{mfr} (experimental or interpolated), by using relationships (1) and (2) presented in Sect. 2. Thus, the corrected Phase-frequency characteristics will follow the patterns shown in Fig. 4.

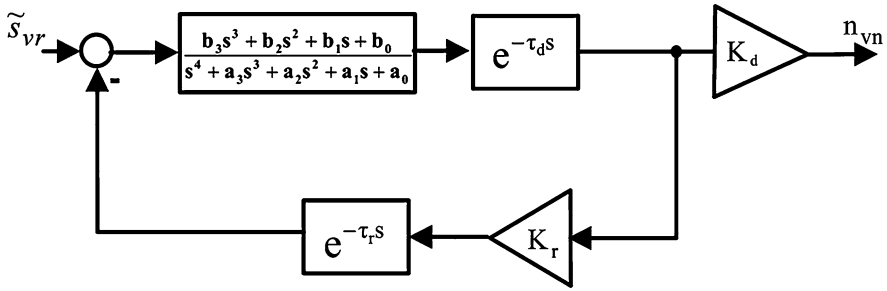


Fig. 9 The general structure of the VN model

3.2 Steps Two and Three of the 3S-M

In the second step of the 3S-M we focused on obtaining parametric models for the VN from the nonparametric models presented in step one. In doing so, we initially used the frequency characteristics for which the uncertainties were not take into account (the IFR signal was computed with method 1) – we’ll consider these frequency characteristics as initial frequency characteristics. This work was presented in detail in [14]. There we arrived at a model with the structure from Fig. 9.

Subsequently, the model was extended in the third step of the 3S-M [15] by adding the p-PI, and the results were validated in the frequency domain (with experimental and interpolated frequency characteristics).

After all this, we continued by trying to develop a VN model from the corrected frequency characteristics (which take into account the uncertainties), by following the same three steps of the 3S-M. Step one is the same as for the first model. In step two, the model structure at which we arrived was simpler – (3). In step three we then extended the model in a manner similar to the one described for the first model, by adding the p-PI, and validating the results in the frequency domain with experimental and interpolated frequency characteristics.

Now let’s compare the results that we have obtained with the two VN complete models (one for the initial frequency characteristics and one for the corrected frequency characteristics).

$$H_{VN}(s) = K \cdot \frac{(b_4s^4 + b_3s^3 + b_2s^2 + b_1s + 1)}{(a_4s^4 + a_3s^3 + a_2s^2 + a_1s + 1)} \tag{3}$$

If until now we have only mentioned the results in the frequency domain, it’s time we also discussed the behavior of the models in the time domain. Here lies the flaw of not taking into account the uncertainties mentioned. Because for the first model the uncertainties were not taken into account, that is the Phase-frequency characteristics were not corrected, the resulting model came out to be unstable (due to the large positive values of the phase on the frequency domain of interest). Thus, time domain simulations are not possible with the first model. On the other hand, when uncertainties are taking into account – for example with corrections of the

Phase-frequency characteristics in this case – the resulting model is stable and thus suitable for time domain simulations also.

4 Conclusions

The paper discusses the main steps in developing a complete adaptive type model of the VN based on experimental frequency characteristics. Due to the uncertainties identified in the data presented in the literature, two actual models were developed for comparison reasons – one in which uncertainties were taken into account and one in which they were ignored. By comparing the results it was concluded that only the model in which the uncertainties were assessed is suitable for time domain simulations (due to issues of instability for the other model). Therefore, in our future work we will further use this model, and as a future development, such a model could be coupled with the output of the vestibular receptors. This will later allow for a complete model of the VS involved in the vestibular-sympathetic reflex to be obtained.

References

1. Carter, J.R., Chester, A.R.: Sympathetic responses to vestibular activation in humans. *Am. J. Physiol. Integr. Comp. Physiol.* **294**, R681–R688 (2008)
2. Radtke, A., Popov, K., Bronstein, A.M., Gresty, M.A.: Vestibulo-autonomic control in man: Short- and long-latency vestibular effects on cardiovascular function. *J. Vest. Res.* **13**(1), 25–37 (2003)
3. Olufsen, M., Ottesen, J., Tran, H., Lipsitz, L., Novak, V.: Modeling baroreflex regulation of heart rate during orthostatic stress. *Am. J. Physiol. Reg. Integr. Comp. Physiol.* **291**, R1355–R1368 (2006)
4. Young, L.R., Meiry, J.L.: A revised dynamic otolith model. *Aerospace Med.* **39**, 606–608 (1968)
5. Oliver, J., Coenen, M.D.: Modeling the vestibulo-ocular reflex and the cerebellum: analytical & computational approaches. PhD Thesis, University of California, San Diego (1999)
6. Roy, A., Iqbal, K.: Kinematic trajectory generation in a neuromusculoskeletal model with somatosensory and vestibular feedback. In: 6th IFAC Symposium on Modeling and Control in Biomedical Systems, Reims, September 2006
7. du Lac S., Lisberger, S.G.: Cellular processing of temporal information in medial vestibular nucleus neurons. *J. Neurosci.*, 8000–8010 (1995)
8. Brodal, P.: *The Central Nervous System: Structure and Function*, 3rd edn. Oxford University Press, London (2004)
9. Raisch, J., Francis, B.A.: *Modeling Deterministic Uncertainty. The Control Handbook*, CRC Press, Boca Raton (1996)
10. Uno, A., Idoux, E., Beranek, M., Vidal, P.-P., Moore, L.E., Wilson, V.J., Vibert, N.: Static and dynamic membrane properties of lateral vestibular nucleus neurons in Guinea pig brain stem slices. *J. Neurophysiol.* **90**, 1689–1703 (2003)
11. Ris, L., Hachemaoui, M., Vibert, N., Godaux, E., Vidal, P.P., Moore, L.E.: Resonance of spike discharge modulation in neurons of the Guinea pig medial vestibular nucleus. *J. Neurophysiol.* **86**, 703–716 (2001)

12. Ramachandran, R., Lisberger, S.G.: Transformation of vestibular signals into motor commands in the vestibuloocular reflex pathways of monkeys. *J. Neurophysiol.* **96**, 1061–1074 (2006)
13. Codrean, A., Ceregan, V., Dragomir, T.-L., Korodi, A.: Interpolative frequency characteristics generators for the vestibular nucleus activity. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 195–199 (2010)
14. Korodi, A., Ceregan, V., Dragomir, T.-L., Codrean, A.: A continuous-time dynamical model for the vestibular nucleus. In: *IFMBE Proceedings, XII Mediterranean Conference on Medical and Biological Engineering and Computing, Chalkidiki*, pp. 627–630 (2010)
15. Ceregan, V., Korodi, A., Dragomir, T.-L., Codrean, A.: An interpolative based dynamical model for the vestibular nucleus. In: *ICCC-CONTI Proc., Timisoara*, pp. 31–36 (2010)

SPECT Lung Delineation

A Complete 3D Approach

Alex Wang and Hong Yan

Abstract This is a review paper of our quest in developing and implementing an automated three-dimensional (3D) lung delineation method capable of handling single photon emission computed tomography (SPECT) lung scans with defective contours and/or varying maximum count value (MCV) and total count value (TCV). Six clinically significant datasets consisting of simulations and real subject scans are used consistently throughout our studies. We first develop a dynamic thresholding method which allows removal of background noise in a 3D volumetric fashion. Next, we implement 3D image processing techniques to enhance the SPECT lung contours. Finally, we develop 3D active contours to perform actual delineation. Quantitative validation using known-volume simulations and qualitative verification via experienced physicians are done to evaluate the methods. We achieve over 90% agreement on average throughout all six datasets.

Keywords SPECT lungs · 3D active contours · Pulmonary embolism

1 Introduction

SPECT has long been favored over CT/MRI in diagnosing pulmonary embolism (PE) for its non-invasiveness and high sensitivity and specificity [1, 2]. Given the criticality of obtaining accurate lung contours for subsequent PE diagnosis [3, 4], common practice is to use a fixed percentage of MCV for thresholding [5–7]. However, the accuracy of this approach is suspect to the presence of localized high deposition of radioactive agents known as “hotspots”. Furthermore, the inherent fuzzy contour and 3D nature of SPECT images also pose a certain degree of com-

A. Wang (✉)

School of Electrical and Information Engineering, The University of Sydney, N.S.W. 2006, Sydney, Australia

e-mail: alexwang80@hotmail.com

plication to traditional planar image processing techniques for contour extraction. This study serves as a review of the journey we have embarked upon in taking on the challenge of developing and implementing an automated method of delineating SPECT lung contours that is completely 3D in nature.

First, through the statistical examination of 3D volumetric information underlying the SPECT images, we develop a dual exponential thresholding (DUET) method that is adaptive in nature [8]. However, the method is limited by the lack of spatial considerations, especially with SPECT lung scans that had defective contours due to low MCV/TCV. This limitation is overcome by combining DUET with traditional planar active contours [9]. While we deliver promising results in a subsequent study where we combine planar image processing techniques with 3D active contours [10], it is the follow-up studies where all aspects of the method is 3D in nature that conclude our quest for a truly 3D approach in delineating SPECT lung contours with great precision [11].

2 Dataset

Two base datasets are repeatedly used throughout our studies to ensure data consistency and comparability of results – 90 SPECT ventilation scans of admitted hospital subjects and 350 Monte Carlo simulations.

Each of the SPECT ventilation scans is acquired using standard protocols. After inhalation of approximately 40 MBq of ^{99m}Tc -Technegas, data is acquired using a dual/triple-head gamma camera fitted with low-energy high-resolution collimators in the format of a 128×128 projection matrix for 120 projection angles. Total acquisition time for the ventilation scan is about eight minutes. The scans are then reconstructed using the OSEM block-iterative algorithm with eight subsets and four iterations. 3D Butterworth low-pass filtering with cut-off frequencies of 0.8 cycles/cm at an order of 9.0 is applied without attenuation correction. The resulting image set contains 128 slices measuring 128×128 pixels in size. Simulation wise, a base set of ten gated projections each consisting of 120 slices measuring 128×128 pixels is generated using Monte Carlo simulation of photon emission from a phantom with a known volume of 61,660 voxels. To reach a statistically significant dataset size, we develop a mass generation method to derive the set of 350 hotspot-free, normally ventilated simulations where each simulation contains 128 slices also measuring 128×128 pixels in size [9]. We also develop a method which adds artificial hotspots into the simulations without impacting the statistical significance of the simulations [11].

Based on these primary SPECT datasets, six subsets shown in Fig. 1 are extracted according to clinically significant image characteristics.

Fig. 1 Dataset characteristics

Dataset #	Dataset Description
1	90 simulations with low MCV/TCV
2	50 simulations with normal MCV/TCV
3	30 simulations with hotspots
4	35 subject scans with low MCV/TCV
5	50 subject scans with normal MCV/TCV
6	25 subject scans with hotspots

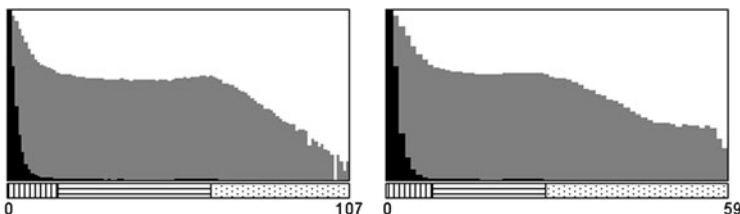


Fig. 2 Sample histograms of real subject SPECT scans. The X and Y axes represent pixel values and corresponding counts. The original values are colored black and the natural logarithmically transformed values are colored grey. The vertical, horizontal, and spotted regions roughly denote background, normal lung tissues, and high-value regions respectively

3 Methods

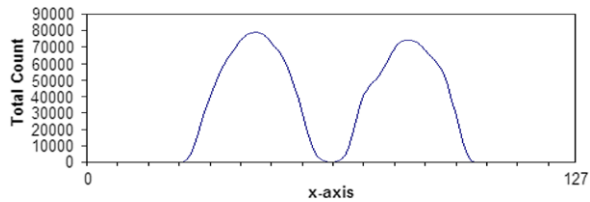
3.1 Duet

As SPECT images contain 3D volumetric information, as well as noisy background and fuzzy contours, existing method of thresholding based on fixed percentages of MCV is found to be inaccurate. Hence, a dynamic adaptation would be more appropriate in performing preliminary background cleansing before application of sophisticated image segmentation methods.

To cater for the 3D nature of the SPECT projections, a summation histogram of all the individual slices in each projection is created. Examination of these histograms reveals triple exponential function like patterns after application of natural logarithmic transformation. There is a rapid component representing the background, a near-horizontal component denoting most of the normally ventilated lung tissues, and a final rapid component signaling the high-value regions. See Fig. 2 for illustration.

Given the above observation, we believe an adaptive threshold may be found as the intercept between the first two exponential functions once appropriate functions are fitted. The data range for exponential function fitting is first established. As our interests lie only in the first two components, a preliminary thresholding is performed to exclude the third component. A horizontal line h on the y -axis which most closely matches the second near-level component is identified via exhaustive linear least squares fitting. Let x_1, x_2, \dots, x_n represent the pixel values, and y_1, y_2, \dots, y_n represent the corresponding histogram counts. Given the search range for h is between zero and y_n , for every incremental horizontal line h_i , the sum of the squared

Fig. 3 Sample horizontal topographic representation of a subject SPECT scan. The dividing point identified in this case is $x = 62$



differences s_i is calculated using Eq. 1, where h_i with the minimal s_i value is defined as h . The preliminary threshold t_{pre} is then defined as the maximum x value with its corresponding y value above h ; only data before t_{pre} is included for further analysis.

$$s_i = \sum_{j=0}^n |h_i - y_j|^2 \quad (1)$$

Once the data range is identified, Eqs. 2.1, 2.2 are fitted to the first two components respectively. As the search ranges for A , B , a , b , k_1 and k_2 are restricted, they are again found via exhaustive linear least squares fitting. Finally, the threshold used for lung delineation is found as the intercept between the two functions.

$$f(x) = Ae^{-ax} + k_1 + h \quad (2.1)$$

$$g(x) = Be^{-bx} + k_2 + h \quad (2.2)$$

3.2 Active Contours

Although we have shown that DUET alone achieves above 90% accuracy for SPECT images that are normally ventilated, its performance is less ideal when the images are low in MCV and/or TCV. Hence, while DUET serves as an excellent noise remover, a more sophisticated and rigorous contour extraction approach is needed. Having assessed various image segmentation methods, we decide upon active contours for its relatively fast processing time and effectiveness in handling fuzzy contours.

Given a SPECT scan, after application of DUET to remove background noise, pre-processing tasks are completed. First, the lungs are divided into the left and the right sides respectively. This is done by creating a horizontal topographic representation of the scan characterized by two peaks representing the two lungs and a trough in between signaling the dividing point [9]. See Fig. 3 for illustration.

Next, the starting region of interest (ROI) for active contours is identified. In our initial planar attempt, ROI is automatically defined as a rectangular box where its boundaries are the first non-zero valued pixels coming in from all four sides [9]. In the subsequent 3D attempt however, such simplistic definition of the starting ROI is not sufficient. Instead, after identifying the rectangular prism containing the

lung, an ellipsoidal mesh made up of triangular strips permuted in a Freudenthal triangulation-like fashion is constructed. Based on this 3D ROI, an $n \times m$ matrix M_m^n is formed where n is the number of points in the ellipsoidal mesh, and m is the number of corresponding immediate neighboring points [11].

Finally, to assist subsequent implementation of active contours, a corresponding gradient image must be created. While standard approach is to apply planar Gaussian smoothing followed by Sobel edge detection on each slice, we apply 3D variations of the Gaussian smoothing $G_{x,y,z}$ and Sobel edge detectors $S_{x,y,z}$ respectively shown in Eqs. 3, 4 to obtain a truly 3D gradient image.

$$G_{x,y,z} = \frac{1}{(2\pi\sqrt{2\pi})\sigma^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \quad (3)$$

$$S_{x,y,z} = S\{\{-1, 0, 1\}, \{-1, 0, 1\}, \{-1, 0, 1\}\} \\ := \left(\begin{array}{l} \left\{ \left[\begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array} \right], \left[\begin{array}{ccc} 2 & 0 & -2 \\ 4 & 0 & -4 \\ 2 & 0 & -2 \end{array} \right], \left[\begin{array}{ccc} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{array} \right] \right\}, \\ \left\{ \left[\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array} \right], \left[\begin{array}{ccc} 2 & 4 & 2 \\ 0 & 0 & 0 \\ -2 & -4 & -2 \end{array} \right], \left[\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array} \right] \right\}, \\ \left\{ \left[\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array} \right], \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right], \left[\begin{array}{ccc} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{array} \right] \right\} \end{array} \right) \quad (4)$$

The basic active contours equation is defined in Eq. 5 where E_{cont} , E_{curv} and E_{img} represent the energy terms of continuity, smoothness and edge attraction respectively, while the three parameters α , β and γ control their degree of sensitivity. The equation can be discretely defined as Eq. 6 where the three energy terms are described in Eqs. 6.1, 6.2, 6.3. The term \bar{d} equals the average distance between the pairs (p_i, p_{i+1}) and ∇I is the spatial gradient of the intensity image I , computed at each point. Given the focus here is to define a 3D approach, we will refer to the three energy terms as surface continuity, surface smoothness, and surface attraction here forth.

$$\varepsilon = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{img}) ds \quad (5)$$

$$\varepsilon = \sum_{i=1}^N \alpha \times E_{cont}^i + \beta \times E_{curv}^i + \gamma \times E_{img}^i \quad (6)$$

$$E_{cont}^i = (\bar{d} - \|p_i - p_{i-1}\|)^2 \quad (6.1)$$

$$E_{curv}^i = \|p_{i-1} - 2p_i + p_{i+1}\|^2 \quad (6.2)$$

$$E_{img}^i = -\|\nabla I\| \quad (6.3)$$

While surface attraction E_{img} remains the same as the traditional planar approach, surface continuity and smoothness have to be modified to cater for 3D implementation. First, with surface continuity E_{cont} , note that the average distance \bar{d} no longer refers to the overall average distance between all contour points and their corresponding neighboring points as in planar active contouring. Instead, it refers to the average distance between the current contour point and its neighboring points in a 3D context, redefining and expanding Eq. 6.1 to become Eqs. 7, 7.1. As for surface smoothness E_{curv} , instead of calculating the squared sum of the distances between a contour point and its immediate pre- or post-neighboring points, the sum of the squared distances between each contour point and all of its neighboring points is calculated, transforming Eq. 6.2 into Eq. 8. By consolidating the observations above, the generic active contours equation shown in Eq. 6 customized for 3D implementation is written as Eq. 9.

$$E_{cont}^i = \sum_{j=1}^m (\bar{d}_i - \|p_i - M_j^i\|) \quad (7)$$

$$\bar{d}_i = \frac{\sum_{j=1}^m \|p_i - M_j^i\|}{m} \quad (7.1)$$

$$E_{curv}^i = \sum_{j=1}^m (p_i - M_j^i)^2 \quad (8)$$

$$\varepsilon = \sum_{i=1}^n \sum_{j=1}^m \alpha \times \left(\frac{\sum_{j=1}^m \|p_i - M_j^i\|}{m} - \|p_i - M_j^i\| \right)^2 + \beta \times \sum_{j=1}^m (p_i - M_j^i)^2 + \gamma \times -\|\nabla I\| \quad (9)$$

As we choose to implement active contours via greedy algorithm for its simplicity and low computational complexity [12], it is crucial to normalize the contribution of each energy term. Hence, for each contour point p_i , surface continuity and smoothness terms are normalized by dividing by the corresponding maximum value, and surface attraction by the norm of the spatial gradient in the neighborhood in which p_i moves as shown in Eqs. 10.1, 10.2, 10.3.

$$norm E_{cont}^i = \frac{E_{cont}^i}{\max E_{cont}^i} \quad (10.1)$$

$$norm E_{curv}^i = \frac{E_{curv}^i}{\max E_{curv}^i} \quad (10.2)$$

$$norm E_{img}^i = \frac{-\|\nabla I\| - \min E_{img}^i}{\max E_{img}^i - \min E_{img}^i} \quad (10.3)$$

While we continue to use the same parameter settings of $\alpha = 1$, $\beta = 1$ and $\gamma = 0.75$ after thorough evaluation against the entire dataset, to prevent the algorithm from entering an endless loop, two constraints are put in place. First, the number

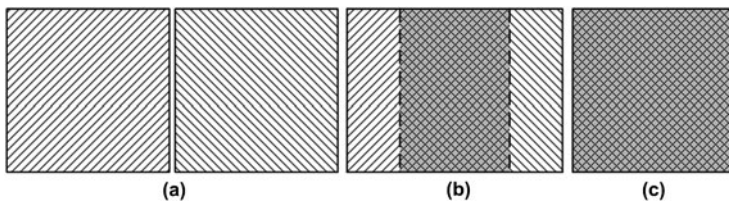


Fig. 4 Illustration of (a) 0%, (b) 50%, and (c) 100% congruency

of iterations I is set to no more than 100. Next, the percentage change in overall average distance between contour points $\Delta \tilde{d}$ must be less than or equal to 0.1% as shown in Eqs. 11, 11.1 such that $\forall a = 1 \dots I$. These constraints are also chosen after evaluating the method against the entire dataset.

$$\Delta \tilde{d} = \frac{\|\tilde{d}_a - \tilde{d}_{a-1}\|}{\max(\tilde{d}_a, \tilde{d}_{a-1})} \tag{11}$$

$$\tilde{d} = \frac{\sum_{i=1}^n \sum_{j=1}^m \|p_i - M_j^i\|}{m \times n} \tag{11.1}$$

4 Implementation and Results

4.1 Evaluation Criteria

To ensure validity and comparability throughout our studies, we develop a measure of congruency expressed in percentages as shown in Eq. 12 where V_d is the delineated volume and V_b is the base comparative volume – 0% means no match whereas 100% represent a perfect match. See Fig. 4 for illustration.

$$C = \frac{V_d \cap V_b}{V_d \cup V_b} \tag{12}$$

The congruency coefficient is performed for each image pair (i.e., the base and the delineated), and the corresponding μ and $\mu \pm 2\sigma$ values for the entire set are then calculated for visualization via Bland-Altman graphs when desired. Given our validation is mainly based on the usage of mean and standard deviation of a dataset’s level of agreement measured in percentages, the results are presented in the form of “mean:standard deviation”, e.g., 100:0% means perfect agreement.

4.2 Simulation Datasets

We first confirm the robustness and accuracy of our methods against the three known-volume simulation datasets previously mentioned: (1) 90 simulations with

(a)	SET 1					SET 2				SET 3		
	Method A	Method B	Method C	Method D	Method E	Method A	Method C	Method D	Method E	Method A	Method D	Method E
$\mu - 2\sigma$	31.79%	72.76%	67.08%	71.98%	60.64%	92.48%	87.29%	91.54%	92.07%	92.34%	92.62%	89.29%
μ	78.00%	88.19%	85.08%	89.83%	81.81%	97.44%	93.83%	96.16%	96.99%	97.37%	96.93%	95.60%
σ	23.11%	7.71%	9.00%	8.92%	10.58%	2.48%	3.27%	2.31%	2.46%	2.51%	2.15%	3.16%
$\mu + 2\sigma$	124.21%	103.61%	103.07%	107.67%	102.97%	102.40%	100.36%	100.78%	101.91%	102.39%	101.23%	101.91%

(b)	SET 4			SET 5			SET 6	
	Method B	Method C	Method D	Method A	Method C	Method D	Method A	Method D
$\mu - 2\sigma$	84.45%	90.24%	90.96%	91.27%	85.43%	87.40%	90.87%	85.95%
μ	95.00%	93.62%	95.15%	96.34%	90.88%	92.85%	96.37%	92.33%
σ	5.27%	1.69%	2.09%	2.53%	2.72%	2.73%	2.75%	3.19%
$\mu + 2\sigma$	105.54%	97.00%	99.33%	101.41%	96.32%	98.31%	101.87%	98.70%

KEY

- SET 1: 90 simulations with low MCV/TCV
- SET 2: 50 simulations with normal MCV/TCV
- SET 3: 30 simulations with hotspots
- SET 4: 35 subject scans with low MCV/TCV
- SET 5: 50 subject scans with normal MCV/TCV
- SET 6: 25 subject scans with hotspots
- Method A: DUET
- Method B: DUET with planar snake
- Method C: DUET with partial 3D snake
- Method D: DUET with full 3D snake
- Method E: Experienced physicians

Fig. 5 Volumetric agreement results in percentages for (a) three sets of simulations, and (b) three sets of real subject SPECT scans

low MCV/TCV, (2) 50 simulations with normal MCV/TCV, and (3) 30 simulations with high MCV due to artificially inserted hotspots. Using the known phantom volume of 61,660 voxels as base volume, congruency is calculated with the delineated volume being the volume extracted by each of our methods. For the set of 50 normal simulations, near-perfect average agreements of 96:2% to 97:3% is obtained across our methods. With the set of 90 low MCV/TCV simulations, an average agreement of 90:9% is achieved using 3D active contours – a significant improvement over 78:23% and 88:8% achieved by DUET alone and planar active contours respectively. Finally, with the set of 30 hotspot-infected simulations, 3D active contours maintain approximately similar mean congruency of 97:2% as DUET alone. See Fig. 5a for details.

4.3 Real Subject Datasets

We now evaluate against the three sets of real subject SPECT scans using results obtained by experienced physicians as the gold standard: (1) 35 scans with low MCV/TCV, (2) 50 scans with normal MCV/TCV, and (3) 25 scans with high MCV due to hotspots. Using results obtained by experienced physicians as the comparative base, average agreements of 95:2% are achieved by our methods for the set of 50 normal scans. For the set of 35 low-count SPECT scans, while we achieve 93:3% mean agreement via 3D active contours – a marginal improvement over previous attempts at planar/partial-3D active contours – performance degradation is observed when comparing against the 96:3% achieved by DUET alone. Finally, with the set of 25 hotspot-infected scans, we achieve yet another highly satisfactory agreement of 92:3% on average. See Fig. 5b for details.

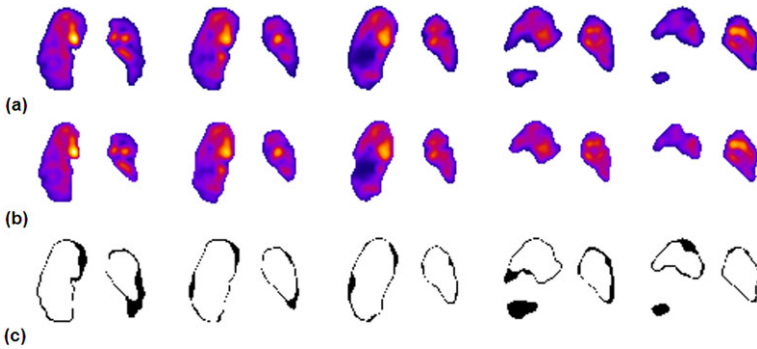


Fig. 6 Sample delineation of a hotspot-infected subject SPECT scan using (a) DUET alone and (b) 3D active contours, with the difference between the two methods shown in (c)

5 Discussion

5.1 Findings

Through our studies, we have shown our success in developing and implementing a SPECT lung delineation method that is truly 3D in nature, achieving an overall agreement of 90% for both simulations and real subject SPECT scans. With low-count SPECT scans, while 3D active contours achieve similar mean agreement of 95% in comparison to the traditional planar approach, it has a lower variance of 2% instead of 5%. It has also improved upon the limitation of over delineation evident in our previous partial attempt at 3D active contours with a marginal increase of 1.5:0.5% in agreement. The results are also consistent with those obtained from simulations with low MCV/TCV. For normally ventilated scans and simulations, the high average agreements of 96:3% and 97:2.5% indicate that DUET alone is sufficient for delineation. Finally, with hotspot-infected subject scans, 3D active contours maintain its ability to accurately delineate lung volumes up to 92:3% mean agreement. While it is evident that 3D active contours yield excellent results in general, it is proven to be especially useful in dealing with low-count scans. Furthermore, due to lack of information on the actual lung volumes of real subject SPECT scans, closer examination of the delineated lung contours of normal and hotspot-infected simulations reveal 3D active contours to be slightly over-aggressive in removing some true peripheral lung tissues. See Fig. 6 for illustration.

5.2 Future Work

Through our progressive studies we have shown the successful development and implementation of a SPECT lung delineation method that is truly 3D in nature. Although additional work is still required to identify the true cause of the over-aggressiveness of the method, examination of Fig. 6 reveal that the method attracts

the contour towards the hotspot edges with higher gradient difference. Furthermore, smaller regions with less prominent voxel values are excluded due to current limitation in segmenting multiple objects. Hence, immediate improvements can be made by incorporating segmentation of multiple objects, and reducing the sensitivity of the method towards hotspot edges. Furthermore, dynamic adjustment of the parameters in active contours may be developed and/or additional energy term(s) introduced. From a dataset perspective, although we have developed fast methods for mass generation of simulations and hotspots, additional phantoms that contain defect(s) similar to those seen in subject scans induced by hotspots and/or other cardiopulmonary disorders are also desired.

6 Conclusion

Although SPECT-CT hybrid machines have been introduced with supporting studies performed on extracting lung contours using anatomical information provided by CT, high total costs is still a major factor that impedes their installation and usage on a wide scale [13, 14]. Based on this notion, we have successfully developed a hybrid method of dynamic thresholding and 3D active contours in delineating SPECT lung contours for PE diagnosis, and the detection of other cardiopulmonary disorders. While the results reveal marginal delineation of true peripheral lung tissues evaluation for scans with normal to high MCV/TCV, overall congruency is still above 90%. The method is most useful in delineating low-count scans with 95% agreement on average. All in all, we have successfully developed a SPECT lung delineation method that is truly 3D in all aspects.

Acknowledgements We would like to thank the physicians and staff at the Royal North Shore Hospital, Australia for the provisioning of the original datasets as well as medical guidance throughout our studies.

References

1. Cross, J.J., Kemp, P.M., Walsh, C.G., et al.: A randomized trial of spiral CT and ventilation perfusion scintigraphy for the diagnosis of pulmonary embolism. *Clin. Radiol.* **53**(3), 177–182 (1998)
2. Howarth, D.M., Lan, L., Thomas, P.A., et al.: 99mTc technegas ventilation and perfusion lung scintigraphy for the diagnosis of pulmonary embolus. *J. Nucl. Med.* **40**(4), 579–584 (1999)
3. Gottschalk, A., Sostman, H.D., Coleman, R.E., et al.: Ventilation-perfusion scintigraphy in the PIOPED study. Part I. Data collection and tabulation. *J. Nucl. Med.* **37**(7), 1109–1118 (1993)
4. Gottschalk, A., Sostman, H.D., Coleman, R.E., et al.: Ventilation-perfusion scintigraphy in the PIOPED study. Part II. Evaluation of the scintigraphic criteria and interpretations. *J. Nucl. Med.* **34**(7), 1119–1126 (1993)
5. Bajc, M., Bitzen, U., Olsson, B., et al.: Lung ventilation/perfusion SPECT in the artificially embolized pig. *J. Nucl. Med.* **43**(5), 640–647 (2002)
6. Bajc, M., Olsson, C., Olsson, B., et al.: Diagnostic evaluation of planar and tomographic ventilation/perfusion lung images in patients with suspected pulmonary emboli. *Clin. Physiol. Funct. Imaging* **24**, 249–256 (2004)

7. Palmer, J., Bitzen, U., Jonson, B., et al.: Comprehensive ventilation/perfusion SPECT. *J. Nucl. Med.* **42**(8), 1288–1294 (2001)
8. Wang, A., Yan, H.: A new automated lung delineation method for SPECT PE diagnosis using adaptive dual exponential thresholding. *Int. J. Imaging Syst. Tech.* **17**(1), 22–27 (2007)
9. Wang, A., Yan, H.: Delineating low-count defective-contour SPECT lung scans for PE diagnosis using adaptive dual exponential thresholding and active contours. *Int. J. Imaging Syst. Tech.* **20**(2), 149–154 (2010)
10. Wang, A., Yan, H.: Delineating SPECT lung contours using 3D snake. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, 17–19 March 2010*, pp. 1494–1499 (2010)
11. Wang, A., Yan, H.: SPECT lung delineation via true 3D active contours. In: *IAENG Transactions on Engineering Technologies*, vol. 5: Special Edition of the International MultiConference of Engineers and Computer Scientists 2010. American Institute of Physics, New York (2010, in press)
12. Trucco, E., Verri, A.: *Introductory Techniques for 3-D Computer Vision*, 1st edn. Prentice Hall, New York (1998)
13. Harris, B., Bailey, D.L., Roach, P.J., et al.: Fusion imaging of computed tomographic pulmonary angiography and SPECT ventilation/perfusion scintigraphy: initial experience and potential benefit. *Eur. J. Nucl. Med. Mol. Imaging* **34**(1), 135–142 (2007)
14. Roach, P.J., Schembri, G.P., Ho Shon, I.A., et al.: SPECT/CT imaging using a spiral CT scanner for anatomical localization: Impact on diagnostic accuracy and reporter confidence in clinical practice. *Nucl. Med. Comm.* **27**(12), 977–987 (2006)